



Calhoun: The NPS Institutional Archive
DSpace Repository

Theses and Dissertations

1. Thesis and Dissertation Collection, all items

1993-12

An analog preprocessing architecture for high-speed analog-to-digital conversion

Esparza, Jorge A.

Monterey, California. Naval Postgraduate School

<http://hdl.handle.net/10945/39681>

This publication is a work of the U.S. Government as defined in Title 17, United States Code, Section 101. Copyright protection is not available for this work in the United States.

Downloaded from NPS Archive: Calhoun



Calhoun is the Naval Postgraduate School's public access digital repository for research materials and institutional publications created by the NPS community. Calhoun is named for Professor of Mathematics Guy K. Calhoun, NPS's first appointed -- and published -- scholarly author.

Dudley Knox Library / Naval Postgraduate School
411 Dyer Road / 1 University Circle
Monterey, California USA 93943

<http://www.nps.edu/library>

2

NAVAL POSTGRADUATE SCHOOL
Monterey, California

AD-A276 737



2



DTIC
ELECTE
MAR 14 1994
S E D

THESIS

**AN ANALOG PREPROCESSING ARCHITECTURE
FOR HIGH-SPEED
ANALOG-TO-DIGITAL CONVERSION**
by

Jorge A. Esparza

December, 1993

Thesis Advisor:
Thesis Co-Advisor:

Phillip E. Pace
Douglas J. Fouts

Approved for public release; distribution is unlimited.

94 3 11 160

94-08207



REPORT DOCUMENTATION PAGE				
1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED			1b. RESTRICTIVE MARKINGS	
2a. SECURITY CLASSIFICATION AUTHORITY			3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited.	
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE			5. MONITORING ORGANIZATION REPORT NUMBER(S)	
4. PERFORMING ORGANIZATION REPORT NUMBER(S)			7a. NAME OF MONITORING ORGANIZATION Naval Postgraduate School	
6a. NAME OF PERFORMING ORGANIZATION Naval Postgraduate School		6b. OFFICE SYMBOL (If applicable) EC		7b. ADDRESS (City, State, and ZIP Code) Monterey, CA 93943-5000
6c. ADDRESS (City, State, and ZIP Code) Monterey, CA 93943-5000		8. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER		
8a. NAME OF FUNDING/SPONSORING ORGANIZATION		8b. OFFICE SYMBOL (If applicable)		10. SOURCE OF FUNDING NUMBERS
8c. ADDRESS (City, State, and ZIP Code)		Program Element No.	Project No.	Task No.
				Work Unit Accession Number
11. TITLE (Include Security Classification) AN ANALOG PREPROCESSING ARCHITECTURE FOR HIGH-SPEED ANALOG-TO-DIGITAL CONVERSION				
12. PERSONAL AUTHOR(S) Esperza, Jorge A.				
13a. TYPE OF REPORT Master's Thesis		13b. TIME COVERED From To		14. DATE OF REPORT (year, month, day) December 1983
15. PAGE COUNT 168				
16. SUPPLEMENTARY NOTATION The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U. S. Government.				
17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)	
FIELD	GROUP	SUBGROUP	Analog-to-Digital Converter; Symmetrical number system; Analog preprocessing for analog-to-digital conversion	
19. ABSTRACT (Continue on reverse if necessary and identify by block number) This thesis investigates the feasibility of implementing an analog-to-digital converter (ADC) based on a new symmetrical number system (SNS). This preprocessing architecture decomposes the analog amplitude analyzing function of an ADC into a number of sub-operations (modules). Each sub-operation folds the analog signal with a folding period proportional to the value of the modules. Through the use of the SNS encoding and recombining the results of the sub-operations, a definitive performance enhancement is achieved. The number of comparators required is reduced considerably, allowing more bandwidth to be used in the folding circuits. The overall design effort demonstrates a 8-bit design with a total of 23 comparators. SPICE simulations are developed and the performance demonstrated. Also identified are the areas in which further research is required.				
20. DISTRIBUTION/AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS REPORT ☐ DTIC USERS			21. ABSTRACT SECURITY CLASSIFICATION Unclassified	
22a. NAME OF RESPONSIBLE INDIVIDUAL Philip E. Pace			22b. TELEPHONE (Include Area Code) (408) 858-3286	
			22c. OFFICE SYMBOL EC/PC	

Approved for public release; distribution is unlimited.

**AN ANALOG PREPROCESSING ARCHITECTURE
FOR HIGH-SPEED ANALOG-TO-DIGITAL CONVERSION**

by

Jorge A. Esparza
Captain, United States Marine Corps
B.S., United States Naval Academy, 1987

Submitted in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

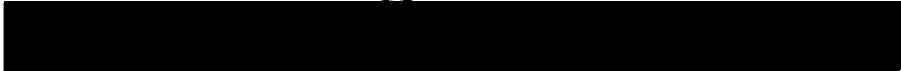
from the

**NAVAL POSTGRADUATE SCHOOL
December 1993**

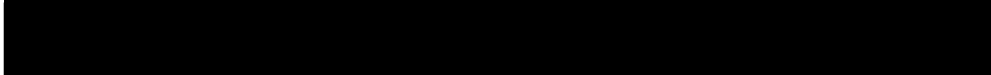
Author:


Jorge A. Esparza

Approved by:


Phillip E. Pace, Thesis Advisor


Douglas J. Fouts, Co-Advisor


Michael A. Morgan, Chairman
Department of Electrical and Computer Engineering

ABSTRACT

This thesis investigates the feasibility of implementing an analog-to-digital converter (ADC) based on a new symmetrical number system (SNS). This preprocessing architecture decomposes the analog amplitude analyzing function of an ADC into a number of sub-operations (moduli). Each sub-operation folds the analog signal with a folding period proportional to the value of the modulus. Through the use of the SNS encoding and recombining the results of the sub-operations, a definitive performance enhancement is achieved. The number of comparators required is reduced considerably, allowing more bandwidth to be used in the folding circuits. The overall design effort demonstrates a 9-bit design with a total of 23 comparators. SPICE simulations are developed and the performance demonstrated. Also identified are the areas in which further research is required.

Accession For	
NTIS CRA&I	☒
DTIC TAB	☐
Unannounced	☐
Justification	
By	
Distribution /	
Availability Codes	
Dist	Avail and/or Special
A-1	

TABLE OF CONTENTS

I. INTRODUCTION	1
II. BACKGROUND	3
A. TYPES OF ANALOG-TO-DIGITAL CONVERTERS	3
B. RESEARCH TRENDS	8
III. ANALOG PREPROCESSING ARCHITECTURE	15
A. GENERAL OVERVIEW	15
B. SYMMETRICAL NUMBER SYSTEM I	15
C. SYMMETRICAL NUMBER SYSTEM II	17
D. SYMMETRICAL NUMBER SYSTEM AND ANALOG PREPROCESSING	19
1. Initial Design Concepts	19
2. Design Modifications	21
IV. DESIGN AND SIMULATION RESULTS	25
A. ANALOG CIRCUIT DESIGN	25
1. Folding Circuit Design and Simulation	25
2. Comparator Design and Simulation	69
B. DIGITAL CIRCUIT DESIGN	76
V. STRENGTHS, LIMITATIONS, AND FUTURE DEVELOPMENTS	100

APPENDIX A SPICE INPUT FILE	102
APPENDIX B INPUT FILE FOR PLA GENERATION	136
APPENDIX C MATLAB PROGRAM TO GENERATE PLA TRUTH TABLE .	147
APPENDIX D DIGITAL SIMULATION OF PLA	151
LIST OF REFERENCES	152
BIBLIOGRAPHY	154
INITIAL DISTRIBUTION LIST	155

LIST OF FIGURES

Figure 2.1: Dual slope ADC.	4
Figure 2.2: Output of integrator during the signal and reference integrate phases.	6
Figure 2.3: Successive Approximation ADC.	7
Figure 2.4: 3-bit flash (parallel) ADC.	9
Figure 2.5: Two-stage ADC with coarse and fine quantization.	11
Figure 2.6: Folding ADC block diagram utilizing coarse and fine quantization.	12
Figure 2.7: Ideal and distorted folded output.	14
Figure 3.1: Block diagram of an analog-to-digital converter with analog preprocessing.	23
Figure 3.2: PLA block diagram.	24
Figure 4.1: A folding circuit.	26
Figure 4.2: Folding circuit DC transfer curve	27
Figure 4.3: Folding circuit's output as the input approaches the value of the power supplies.	28
Figure 4.4: Periodicity in the folding circuit's output.	30
Figure 4.5: Modulus 23 folding circuit block diagram.	31
Figure 4.6: Modulus 23 folding circuit schematic.	32

Figure 4.7:	Comparator outputs of simulated mod 23 folding circuit	35
Figure 4.8:	Mod 23 folding circuit output from a transient analysis at 10 Hz.	36
Figure 4.9:	Mod 23 folding circuit output from a transient analysis at 1 KHz.	37
Figure 4.10:	Mod 23 folding circuit transient analysis at 10 Khz.	38
Figure 4.11:	DC transfer curve of one fold with a 0.05 mA current source.	39
Figure 4.12:	DC transfer curve of one fold with a current source of 0.2 mA.	40
Figure 4.13:	Narrow fold width due to small current source.	42
Figure 4.14:	Wide fold width due to larger current source.	43
Figure 4.15:	Loading problem in folding circuits. . . .	44
Figure 4.16:	Voltage shifter circuit.	45
Figure 4.17:	Revised modulus folding circuit.	46
Figure 4.18:	Fold height inconsistency in modulus folding circuit.	48
Figure 4.19:	Fold height inconsistency variations. . .	51
Figure 4.20:	Bottom of folds from Figure 4.19. . . .	52
Figure 4.21:	Fold height inconsistency test part 1. .	53
Figure 4.22:	Fold height inconsistency test part 2. .	54
Figure 4.23:	Modulus 20 folding circuit output. . . .	59

Figure 4.24:	Modulus 7 folding circuit output.	61
Figure 4.25:	Modulus 8 folding circuit output.	62
Figure 4.26:	Modulus 11 folding circuits output.	63
Figure 4.27:	Modulus 7 comparator levels.	64
Figure 4.28:	Modulus 8 comparator levels.	65
Figure 4.29:	Modulus 11 comparator levels.	66
Figure 4.30:	Clocked comparator circuit diagram.	70
Figure 4.31:	Clocked comparator's output when threshold voltage is greater than the input voltage.	71
Figure 4.32:	Clocked comparator's output when the threshold voltage is less than the input voltage.	73
Figure 4.33:	Clocked comparator output with oscillations.	74
Figure 4.34:	Static comparator with transmission gate.	75
Figure 4.35:	Operation of a transmission gate.	77
Figure 4.36:	Mod 7 comparator outputs.	78
Figure 4.37:	Mod 7 comparator and 1st decoder outputs.	79
Figure 4.38:	Mod 7 1st and 2nd decoder outputs.	80
Figure 4.39:	Mod 8 comparator outputs.	81
Figure 4.40:	Mod 8 comparator and 1st decoder outputs.	82
Figure 4.41:	Mod 8 1st decoder outputs.	83
Figure 4.42:	Mod 8 2nd decoder output.	84
Figure 4.43:	Mod 11 comparator outputs.	85
Figure 4.44:	Mod 11 comparator outputs continued.	86
Figure 4.45:	Mod 11 1st decoder output.	87
Figure 4.46:	Mod 11 1st decoder output continued.	88

Figure 4.47:	Mod 11 2nd decoder output.	89
Figure 4.48:	First half of the mod 7 decoder.	91
Figure 4.49:	CMOS gates used in decoder design.	92
Figure 4.50:	1st decoder simulation showing the input comparator levels.	93
Figure 4.51:	1st decoder simulation showing decoder output.	94
Figure 4.52:	Second half of the mod 7 decoder.	96
Figure 4.53:	2nd decoder simulation showing the decoder output.	97
Figure 4.54:	Partial layout of a PLA.	98

ACKNOWLEDGEMENT

I would like to thank the Marine Corps for giving me the opportunity to be intellectually challenged at the Naval Postgraduate School.

I would like to express my sincere appreciation to Professor P.E. Pace, my thesis advisor, for his encouragement and guidance, and to Professor D.J. Fouts, my thesis co-advisor, for his assistance through many hours of computer simulations.

Last but not least, I want to thank my wife, Tracy, my son, Antonio, and my daughter, GiGi, for their support, patience, and moral support.

I. INTRODUCTION

This thesis begins with a discussion on the theory of analog-to-digital converters. The discussion progresses from the slowest type of ADC's, like the dual slope, to the fastest type which is the flash or parallel converter. Advantages and disadvantages are discussed for each type of converter, but current research trends have been to improve on the flash type converter. One of these improvements is the basis of this thesis.

This thesis proposes a new analog preprocessing architecture for analog-to-digital converters (ADC) which is based on a symmetrical number system (SNS). The number system lends itself favorably to the folding circuit methodology which is used in the design that is presented here and in other designs which have been presented by other researchers. Utilizing both the symmetrical number system and folding circuits, this thesis will demonstrate a considerable decrease in the number of comparators that are used in the quantization process. The number of comparators is one of the major factors driving flash analog-to-digital converter in terms of speed, size, and power consumption.

After the background discussion on the theory of analog-to-digital converters, the mathematical basis for the symmetrical number system is reviewed. Additionally, a

discussion on how this number system can be incorporated into an analog preprocessing architecture to improve flash type ADC's ensues. The bulk of this thesis documents the research conducted in the design and simulation of the analog folding circuits utilizing a 2 micron CMOS process. There were many challenges and obstacles which presented itself in this research effort such as frequency response and loading problems, but most of these were resolved.

After the analog portion of the ADC design is accomplished, the digital design of the ADC is presented. The design and simulation of the decoder and the PLA which are used in the design are quite straight forward with no major problem encountered. This thesis discusses the digital logic involved in the operation of the decoder in terms of truth tables and simulations, and the purpose and functionality of the programmable logic array (PLA) is reviewed with simulation used as validation.

Finally, this thesis ends with a recapitulation of strengths and problems of the design presented in this thesis. Also, this thesis proposes possible solutions to the pending problems and improvements for further design work. Future research areas based on this thesis are also examined.

II. BACKGROUND

A. TYPES OF ANALOG-TO-DIGITAL CONVERTERS

Analog-to Digital converters accept an analog signal as an input and convert it into a digital form as an output. Computers use these devices to sample information from its external environment and manipulate the data as may be necessary to perform a useful function. Depending on the conversion rates of the ADC, the device can be used to process a variety of analog signals from slow moving voltage signals to faster video signals. There are generally three different types of ADC's: dual slope, successive approximation, and flash (or parallel). Dual slope ADC's have the slowest conversion rates, approximately 300 ms. The conversion process is divided into two phases. The first phase is the signal integrate phase where an input signal, V_{in} , is applied to an integrator (see Figure 2.1). The output voltage of the integrator, V_{out} , ramps with a slope proportional to V_{in} and ceases after a predetermined time has transpired. The next phase is the reference integrate phase, which where the analog-to-digital conversion takes place. At this point, the integrator is disconnected from V_{in} and connected to the reference voltage, V_{ref} . Concurrently, a counter begins

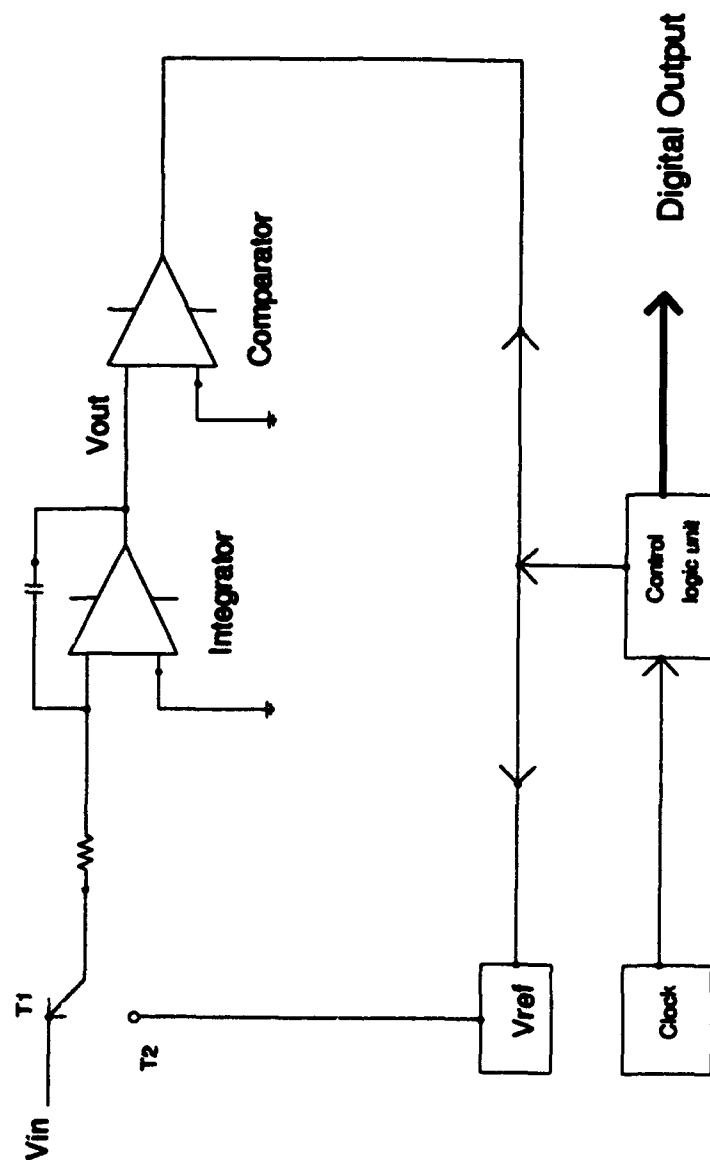


Figure 2.1: Dual slope ADC.

to count at a specified rate, f_c . V_{out} ramps at a constant rate in the opposite direction of V_{in} from the signal integrate phase. Once V_{out} reaches 0.0V, the comparator tells the control logic to terminate this phase and stop the counter. As can be seen in Figure 2.2, T_2 is proportional to V_{out} which in turn is proportional to V_{in} :

$$T_2 = T_1 \frac{V_{in}}{V_{ref}}. \quad (2.1)$$

The digital output is contained in the contents of the counter and may be derived using the following equation:

$$D_{out} = f_c T_1 \frac{V_{in}}{V_{ref}} \quad (2.2)$$

where D_{out} is the digital output, f_c is the frequency of the clock driving the counter [Refs. 1-4].

A faster type of analog-to-digital converter is the successive approximation ADC. These ADC's have conversion times of several microseconds and are used primarily to digitize audio signals. The conversion begins when a start conversion signal is applied (See Figure 2.3). The successive approximation register, SAR, applies a digital signal to a digital-to-analog converter (DAC) with the most significant bit (MSB) set and the other bits cleared. The DAC produces an analog output, V_o , which is applied to a comparator. The

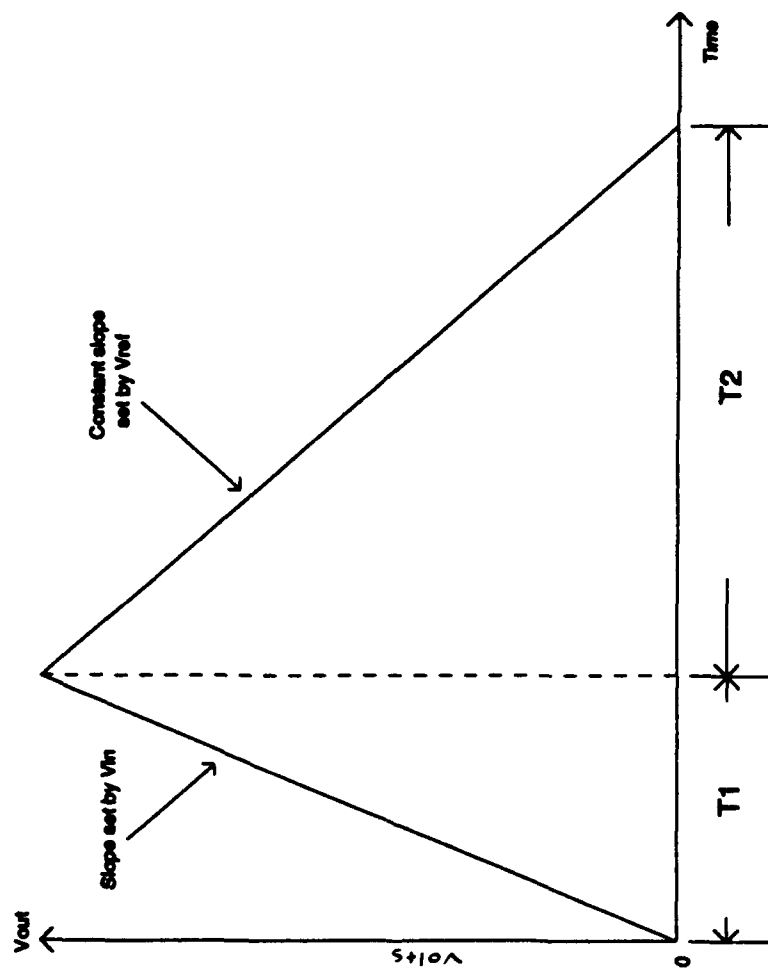


Figure 2.2: Output of integrator during the signal and reference integrate phases.

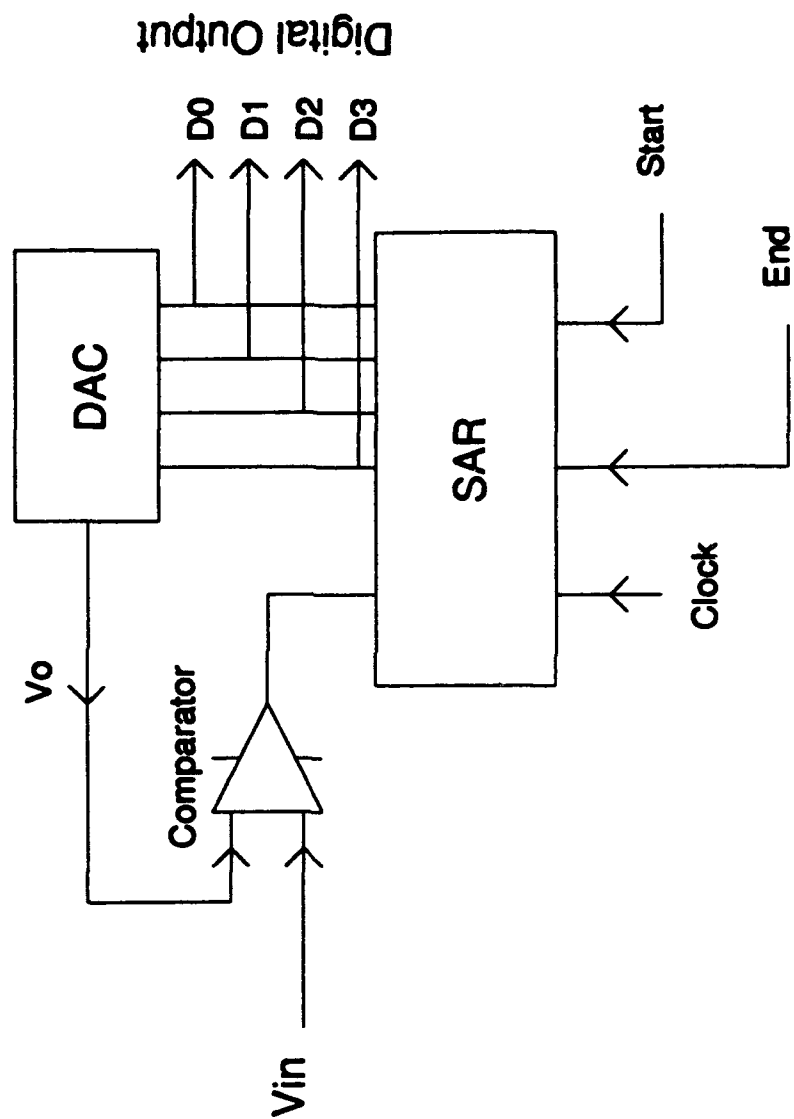


Figure 2.3: Successive Approximation ADC.

comparator compares V_0 and the analog input signal, V_{in} . If $V_{in} > V_0$, the bit being resolved remains set. If $V_{in} < V_0$, then the bit being resolved is cleared. Once the bit is resolved, the conversion cycle is considered complete. The SAR will then set the next lower significant bit which begins a new conversion cycle. When the last bit has been converted, an end of conversion signal is produced, indicating that the digital output is valid [Refs. 1-4].

The fastest type of analog-to-digital converter is the Flash or Parallel ADC. These ADC's are generally used for video signals because conversion rates are in the hundreds of microseconds. With this type of ADC, an input voltage signal is applied to a parallel bank of 2^n-1 comparators which compares the input signal to 2^n-1 reference voltages, where n is the number of bits in the resolution (See Figure 2.4). To realize a digital representation of the input signal, combinational logic is utilized to convert the output of the comparators to a standard binary format [Refs. 1-4].

B. RESEARCH TRENDS

Current research has endeavored to retain aspects of the Flash ADC architecture due to its high-speed characteristic. Also, there exists a need to reduce chip area and power consumption. Current trends have suggested reducing the number of comparators to address the chip area and power problem. One area of research is in optimizing the coarse-

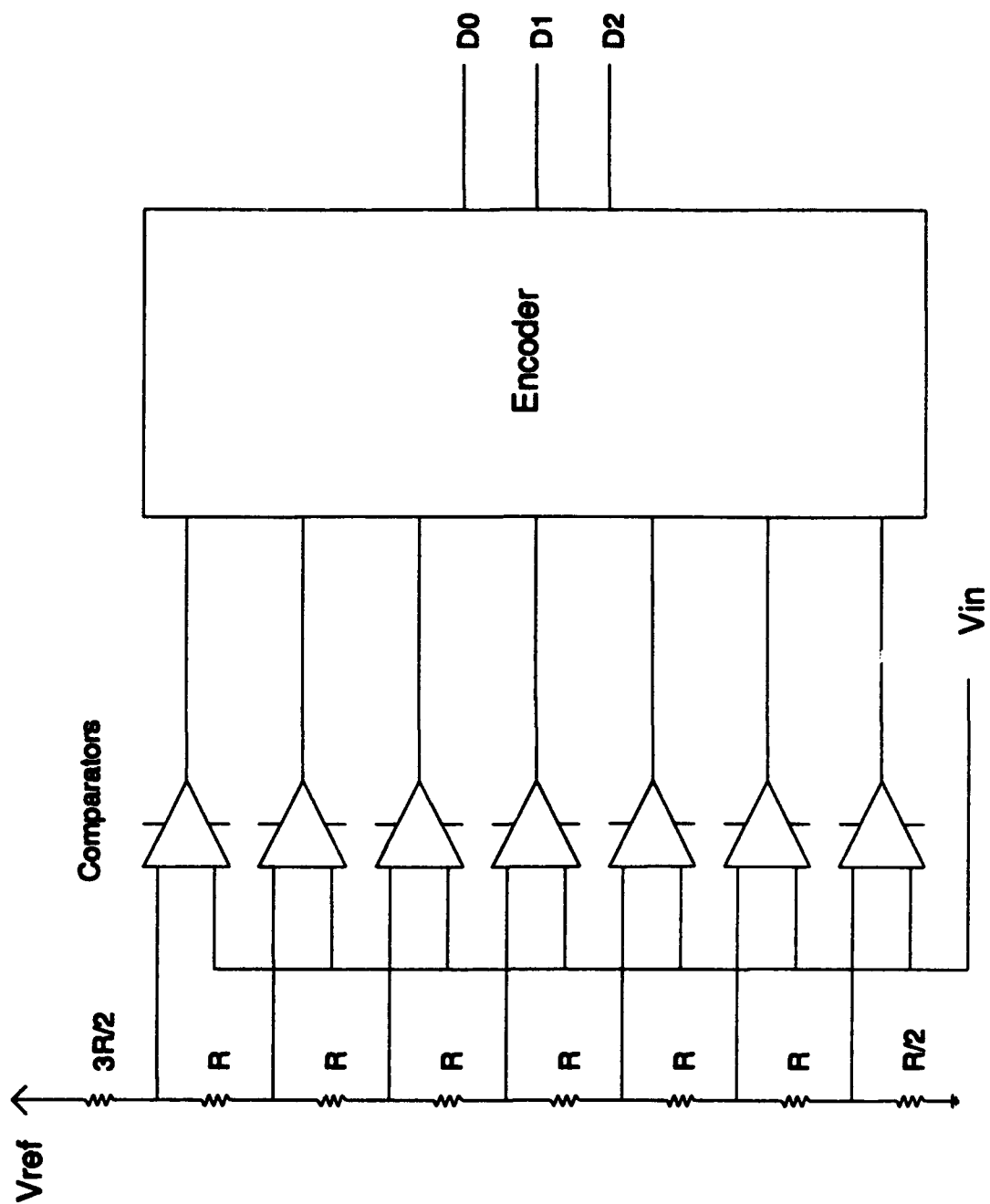


Figure 2.4: 3-bit flash (parallel) ADC.

fine quantization approach. In this approach, the input signal is applied to a coarse analog-to-digital converter to resolve the most significant bits (MSB's). These MSB's are then converted into an analog form utilizing a digital-to-analog converter. This analog value is then subtracted from the input signal and the result is applied to a second analog-to-digital converter to resolve the remaining least significant bits (LSB's). This approach dramatically reduces the number of comparators used in the quantization process. For example, in a pure flash ADC, 255 comparators would be needed to resolved 8-bits. However, in the course-fine quantization using a 3-bit flash comparator for the course and a 5-bit flash comparator for the fine quantization only 40 would comparators would be needed (see Figure 2.5). The drawback to this approach is the need to incorporate a digital-to-analog converter which introduces a timing problem [Refs. 5-6].

Another method in reducing the number of comparators in a flash comparator configuration is the use of folding analog preprocessing [Refs. 7,8]. In this process shown in Figure 2.6, the input signal goes through a course quantization using flash ADC to resolve the MSB's. Simultaneously, the input signal is transformed into an approximately triangular waveform. This waveform is then quantized using a flash ADC to obtain the LSB's. The number of comparators used in this

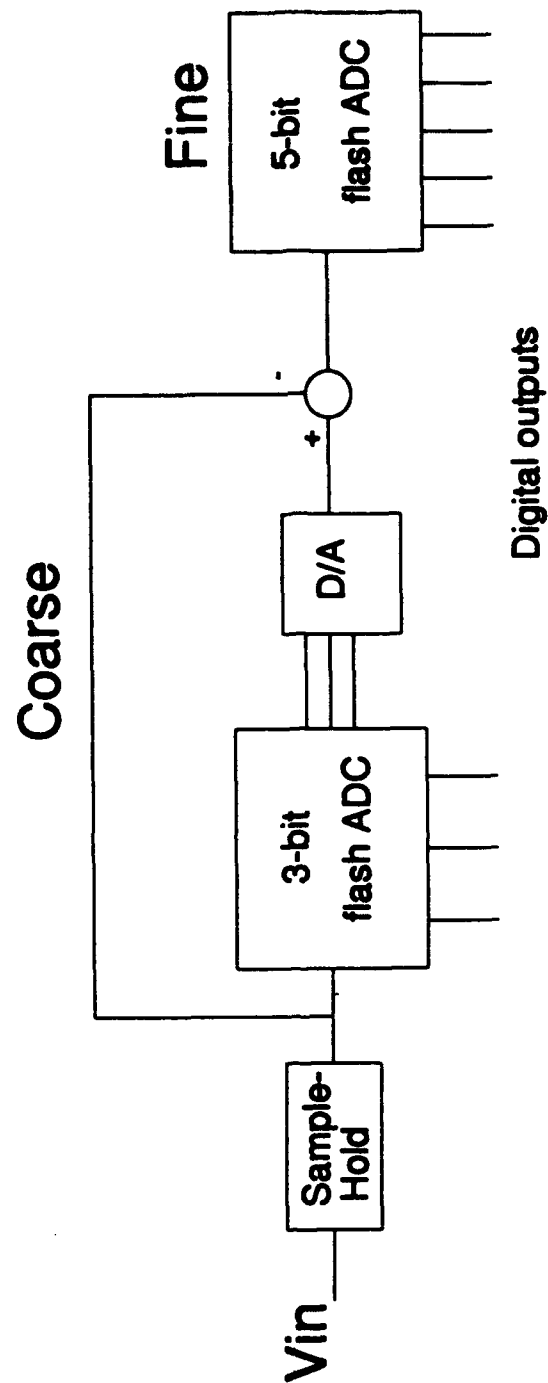


Figure 2.5: Two-stage ADC with coarse and fine quantization.

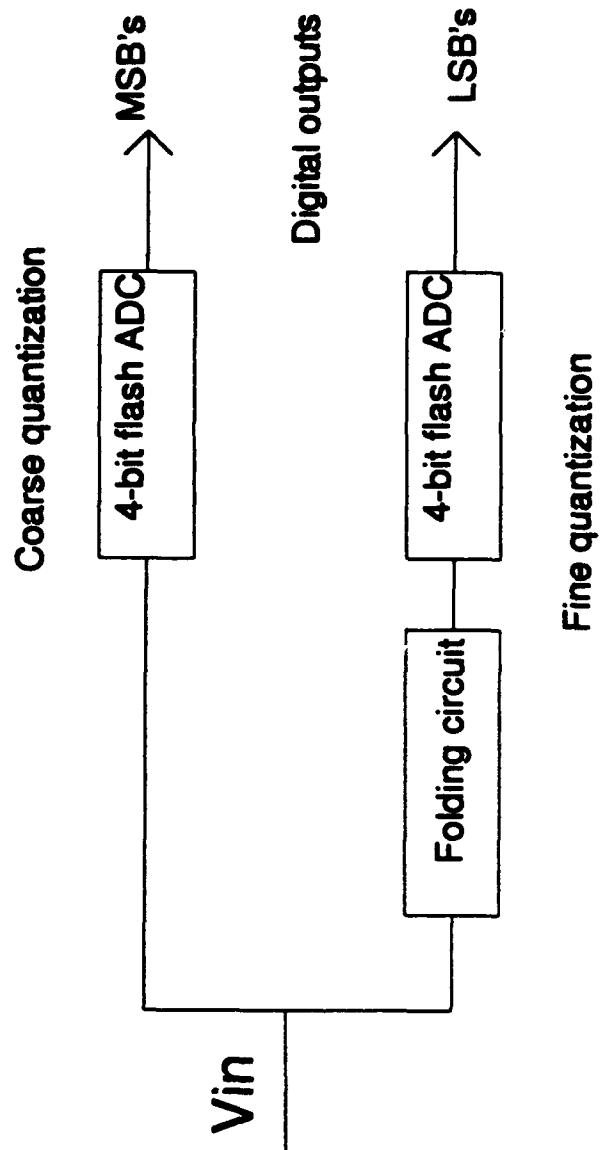


Figure 2.6: Folding ADC block diagram utilizing coarse and fine quantization.

folding technique is less than that of the course-fine quantization technique. This is very desirable but the drawback is the reduction of resolution of input signals at high frequencies because the tips of the triangular waveform tend to round off [Ref. 6] (see Figure 2.7). These round off regions in the waveform introduce errors in the conversion process. One method to resolve this problem is to use a double folding circuit. In this method, when one of the outputs of the double folding circuit moves into a rounded region, the other output will take over as it comes into its linear region [Ref. 9].

Incorporating folding and interpolation techniques to ADC design is another research trend which seeks to reduce the required number of comparators [Refs. 5,10,11]. In this technique, a certain number of folding-amplifier blocks are omitted. The output signals of the omitted folding-amplifier blocks are reconstructed from the remaining folding-amplifier blocks. The omission of these folding blocks directly translates into the reduction of the number of comparators. In one 8-bit implementation the number of required comparators was 64 [Ref. 10].

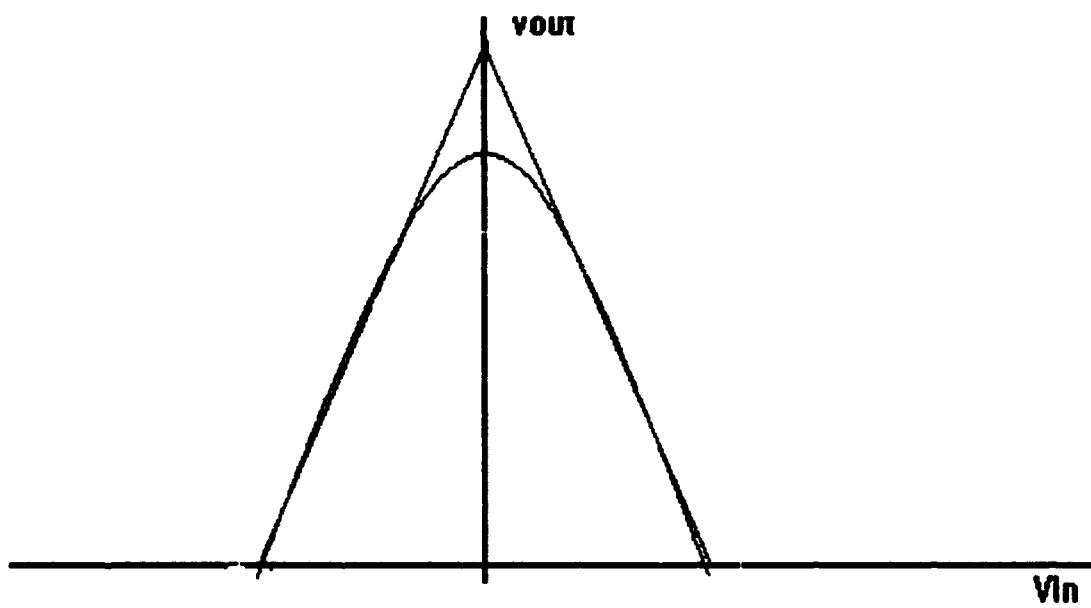


Figure 2.7: Ideal and distorted folded output.

III. ANALOG PREPROCESSING ARCHITECTURE

A. GENERAL OVERVIEW

This thesis seeks to modify the basic flash ADC configuration by incorporating a folding analog preprocessing architecture before the signal is quantized by a parallel set of comparators. Through the use of the symmetrical number system (SNS) as discussed below, the preprocessing architecture decomposes the analog input signal into sub-operations or moduli. Within each modulus circuit, the analog input signal folds with a period corresponding to the value corresponding to the modulus. In effect, this system reduces the preprocessing problem to a set of simpler, parallel operations. The result of these compartmentalized operations are combined and later digitally converted. This digital conversion produces an output which represents the input signal in the SNS format and later to the standard binary format.

B. SYMMETRICAL NUMBER SYSTEM I

The first version of the symmetrical number system (SNS I) used in the initial design is composed of a set of pairwise relatively prime moduli m_i [Ref. 12]. Each modulus is formed by a unique set of integer values. This series of integer values repeats itself after the number of elements within a

modulus set equals the value of the modulus. In effect, the modulus generates a symmetrical waveform which fold with a period equal to the value of the modulus. The equation that generates an individual odd modulus set is

$$X_m = [0, 1, \dots, \lfloor \frac{m}{2} \rfloor, \lfloor \frac{m}{2} \rfloor, \dots, 1] \quad (3.1)$$

where the function $\lfloor x \rfloor$ is the greatest integer less than or equal to x . For an individual even modulus set the equation is

$$X_m = [0, 1, \dots, \frac{m}{2}, \frac{m}{2}-1, \dots, 1] \quad (3.2)$$

Using these equation, a modulus 5 (mod 5) set would be $\{0, 1, 2, 2, 1, 0, 1, 2, \dots\}$.

A combination of these moduli describe a number system which has a one-to-one correspondence with the residue system up to a certain range. The numerical patterns produced by the merging of a set of moduli will not repeat until a dynamic range has been reached. The dynamic range for a set of moduli where one modulus is even is

$$\hat{M} = \min \left\{ \frac{m_{\text{even}}}{2} \prod_{l=2}^j m_{i_l} + \prod_{l=j+1}^N m_{i_l} \right\} \quad (3.3)$$

where N is the number of moduli, j ranges from 2 to $N-1$ and the subscripts i , run over all permutations $\{2, 3, \dots, N\}$.

When the set of moduli are all odd, the dynamic range is given by

$$\hat{M} = \min\left\{\frac{1}{2} \prod_{l=1}^j m_{i_l} + \frac{1}{2} \prod_{l=j+1}^N m_{i_l}\right\} \quad (3.4)$$

where j ranges from 1 to $N-1$ and the subscripts i_l run over all permutations $\{1, 2, \dots, N\}$. From the two above equations, the dynamic range would be larger for a set of moduli containing an even modulus than for a set of all odd moduli where the even modulus is increased by one. For example, the dynamic range for the set of moduli, $m_1=3$, $m_2=5$, $m_3=7$, is calculated to be $M=3.5+7.5 = 11$. When $m_1=3$ is changed to $m_1=2$ then the dynamic range increases to 12, and is illustrated in Table 3.1. Thus, a greater efficiency is obtained when an even modulus is used in combination with odd moduli.

C. SYMMETRICAL NUMBER SYSTEM II

The second version of the symmetrical number system (SNS II) used in the final design is composed of a set of pairwise relatively prime moduli m_i , as in the SNS I. Each modulus is formed by a unique set of integer values. This series of integer values repeats itself after the number of elements within each modulus set is equal to twice the value of the modulus. As in the SNS I, the modulus generates a symmetrical waveform, but the fold period of the waveform is equal to twice the value of the modulus. The equation that generates an individual modulus set is

**TABLE 3.1 TWO SNS SYSTEMS
SHOWING THE INCREASED EFFICIENCY
OF AN EVEN MODULUS**

SNS MODULI							
Dynamic Range	3	5	7		2	5	7
0	0	0	0		0	0	0
1	1	1	1		1	1	1
2	1	2	2		0	2	2
3	0	2	3		1	2	3
4	1	1	3		0	1	3
5	1	0	2		1	0	2
6	0	1	1		0	1	1
7	1	2	0		1	2	0
8	1	2	1		0	2	1
9	0	1	2		1	1	2
10	1	0	3		0	0	3
11	1	1	3		1	1	3
12	0	2	2		0	2	2

$$x_m = [0, 1, \dots, m-1, m-1, \dots, 1, 0]. \quad (3.5)$$

Using this equation, a modulus 4 (mod 4) set would be (0,1,2,3,3,2,1,0,...). Similar to the SNS I, the system based on a set of moduli will produce SNS vectors which will not repeat until the dynamic range has been reached. The dynamic range M for a given set of SNS II moduli is given by:

$$M = \prod_{i=1}^N m_i. \quad (3.6)$$

For example, if the system is defined by the moduli, $m_1=3$ and $m_2=4$, the dynamic range M would be equal to 12 as is shown in Table 3.2 [Ref. 13].

D. SYMMETRICAL NUMBER SYSTEM AND ANALOG PREPROCESSING

1. Initial Design Concepts

This thesis endeavored to incorporate the symmetrical number system as the basis for the design of an analog preprocessing for an analog-to-digital converter. In this design, an input signal is applied in parallel to an appropriate number of modulus folding circuits as shown in Figure 3.1. The folding circuits will fold the input signal with a period based on the value of the respective modulus. The output of each fold is then quantized by a bank of comparators connected in parallel. At this point, the number of comparators that are high represents the value of the input signal in the SNS format.

TABLE 3.2 DYNAMIC RANGE OF AN
SNS II SYSTEM

Dynamic Range	SNS II Moduli	
	4	3
0	0	0
1	1	1
2	2	2
3	3	2
4	3	1
5	2	0
6	1	0
7	0	1
8	0	2
9	1	2
10	2	1
11	3	0
12	3	0

After the input signal has been preprocessed and quantized, the outputs of the comparators are applied to a decoder. This decoder transforms the comparator output values into a binary thermometer format, and then to a straight binary number. This binary vector represents the highest comparator level reached by the input signal within each modulus. When combined, the output of each of the modulus decoders represents the input value in its SNS binary format. Since the SNS binary format is not easily recognizable, a programmable logic array (PLA) is used to convert a SNS binary number to a standard binary number (see Figures 3.1 and 3.2).

2. Design Modifications

Although the initial design of the analog preprocessing architecture was based on the first version of the symmetrical number system, a second version of the symmetrical number system caused slight modifications of the initial design. In this system, the dynamic range, M , is simply the product of the values of the moduli. For the moduli used in this thesis, $m_1=7$, $m_2=8$, and $m_3=11$, the dynamic range is calculated to be $M=7 \times 8 \times 11=616$, which is well above 512 which is needed for a 9-bit resolution. The decision to change number systems was based on the benefits realized from the newer version. First, the new version reduces the number of moduli required to generate a specific dynamic range. In

the old design, four moduli were required to generate a dynamic range large enough to be represented by nine bits. The SNS II only requires three moduli. Second, because of the reduction of the number of moduli, the total number of comparators utilized is also reduced. Also, within each modulus of the new design, the number of comparators is less than the older design. In the first design, the number of comparator per modulus, m_i , is $[m_i/2]$. But, in the second design, the values of each modulus are considerably less than the moduli used in the first design for the same dynamic range. Because of this, the number of comparators for the second design, m_i-1 , will be less than those contained in the first design.

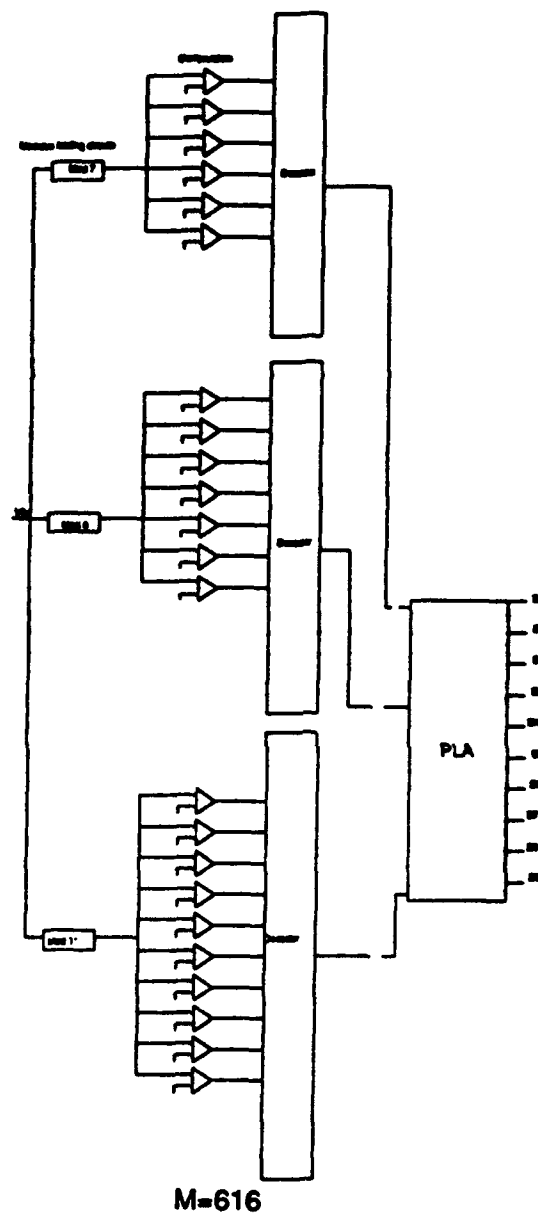
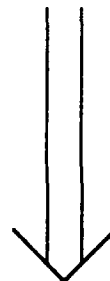
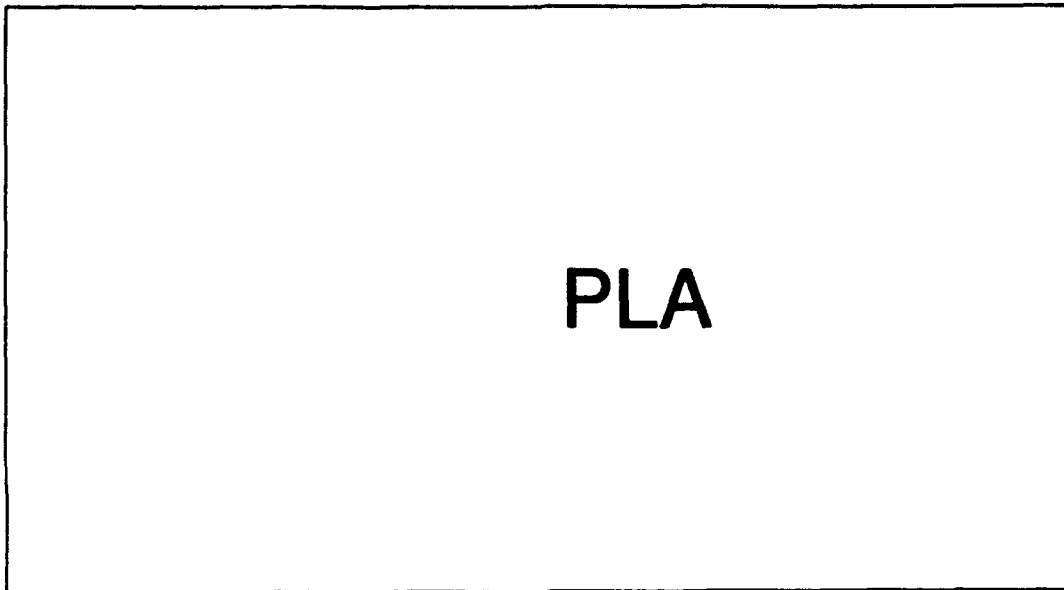
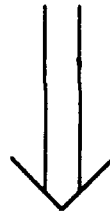


Figure 3.1: Block diagram of an analog-to-digital converter with analog preprocessing.

SNS Representation



Binary Representation

Figure 3.2: PLA block diagram.

IV. DESIGN AND SIMULATION RESULTS

A. ANALOG CIRCUIT DESIGN

1. Folding Circuit Design and Simulation

The first step in the design of the analog preprocessing architecture is the design of the folding circuit. The folding circuit as shown in Figure 4.1 is composed of a pair of MOS differential amplifiers. An input signal (V_{in}) is applied to the gate of M3 of the lower differential amplifier, and a reference voltage (V_{ref}) is applied to the gate of M4. The output (V_{out}) of the folding circuit is taken from the source of M1 and M2. The resulting transfer function is depicted in Figure 4.2. This relationship is achieved because the lower differential amplifier consisting of M3 and M4 provide the straight-line section symmetrical about V_{ref} , while the upper differential amplifier consisting of M1 and M2 provide an analog OR function. This function transfers the higher of the base voltages of M1 and M2 to the output V_{out} .

Spice simulations of the folding circuit demonstrated that as the value of V_{ref} reached the value of the power supplies driving the circuit, the folded output, V_{out} , became distorted as shown in Figure 4.3. This indicated that the value of the final reference voltage, and thus the dynamic

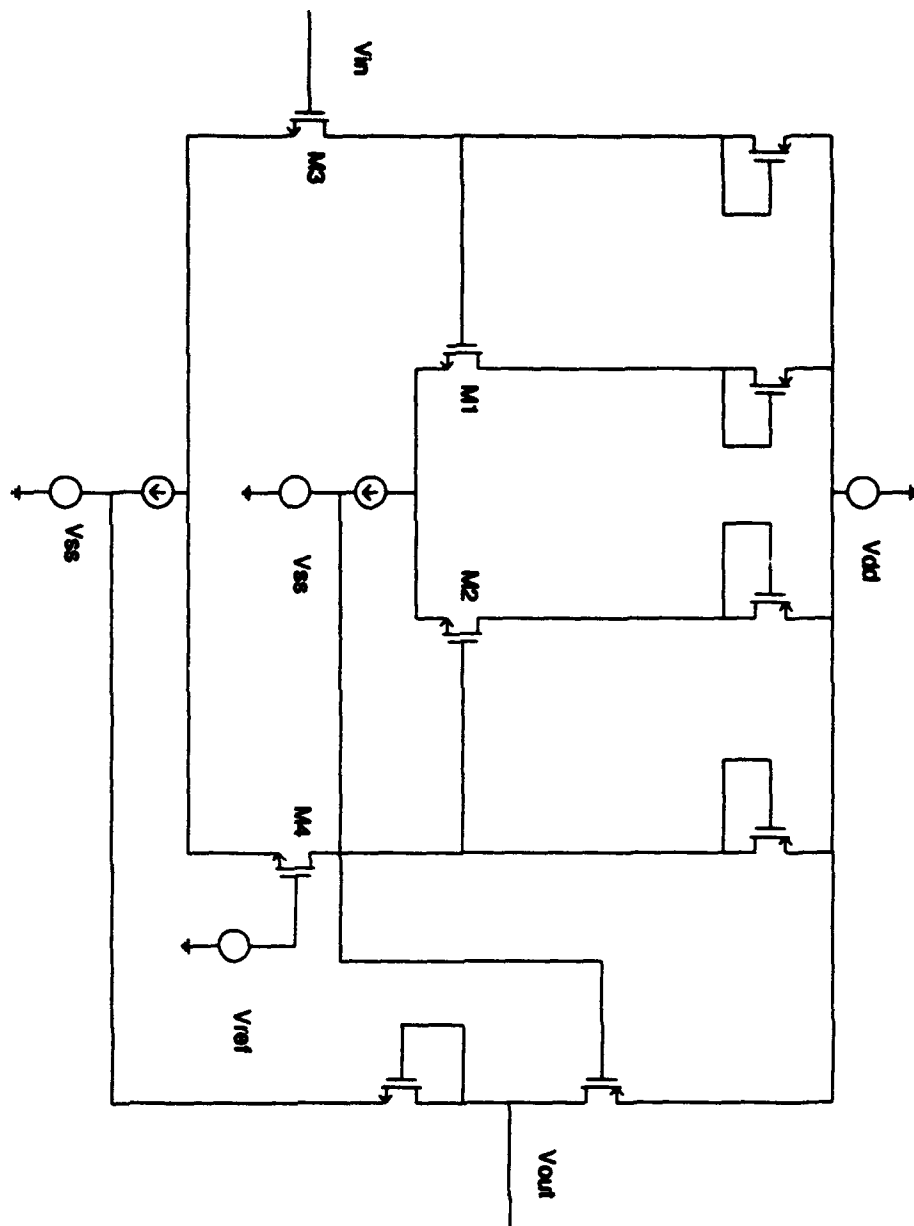


Figure 4.1: A folding circuit.

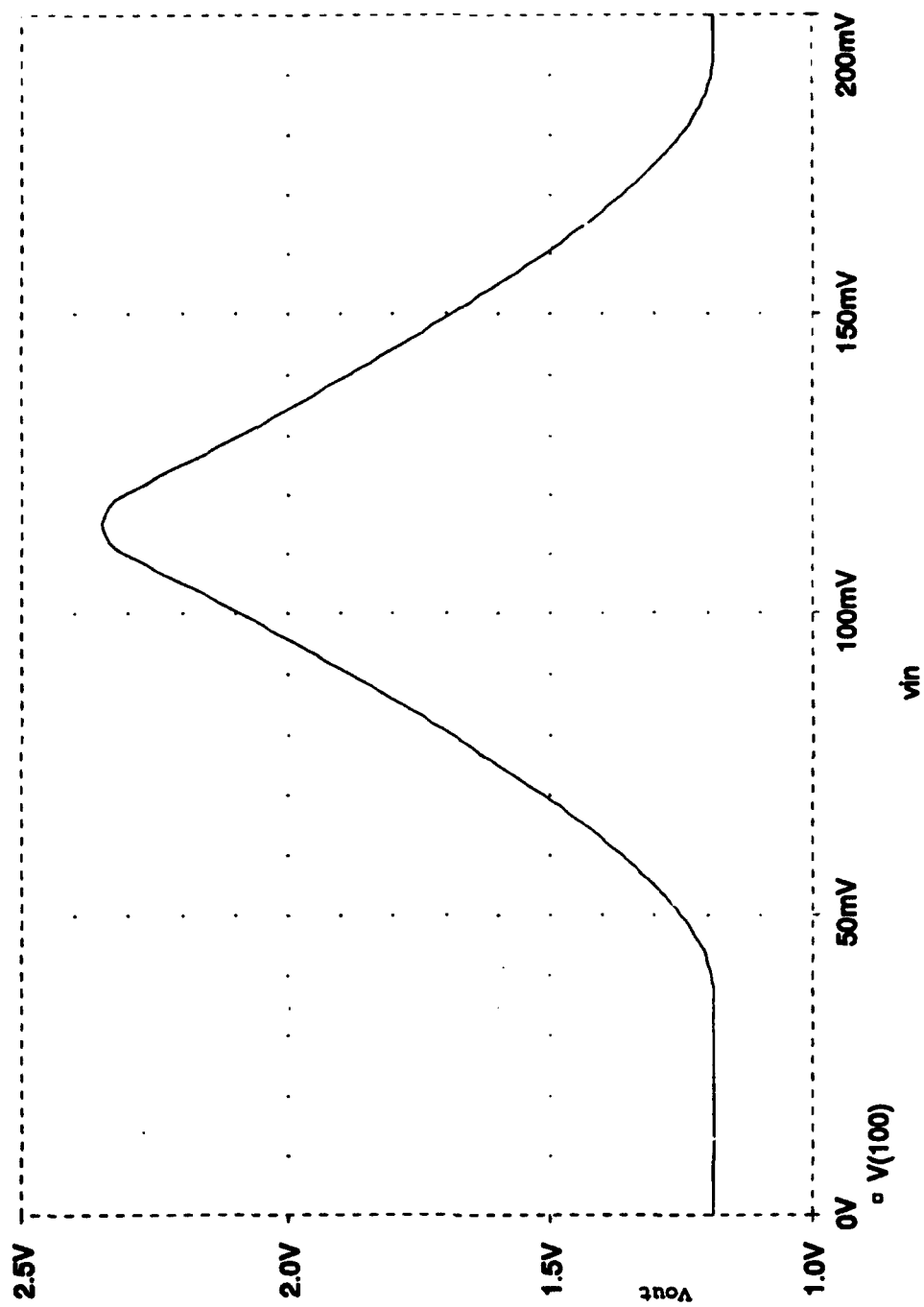


Figure 4.2: Folding circuit DC transfer curve.

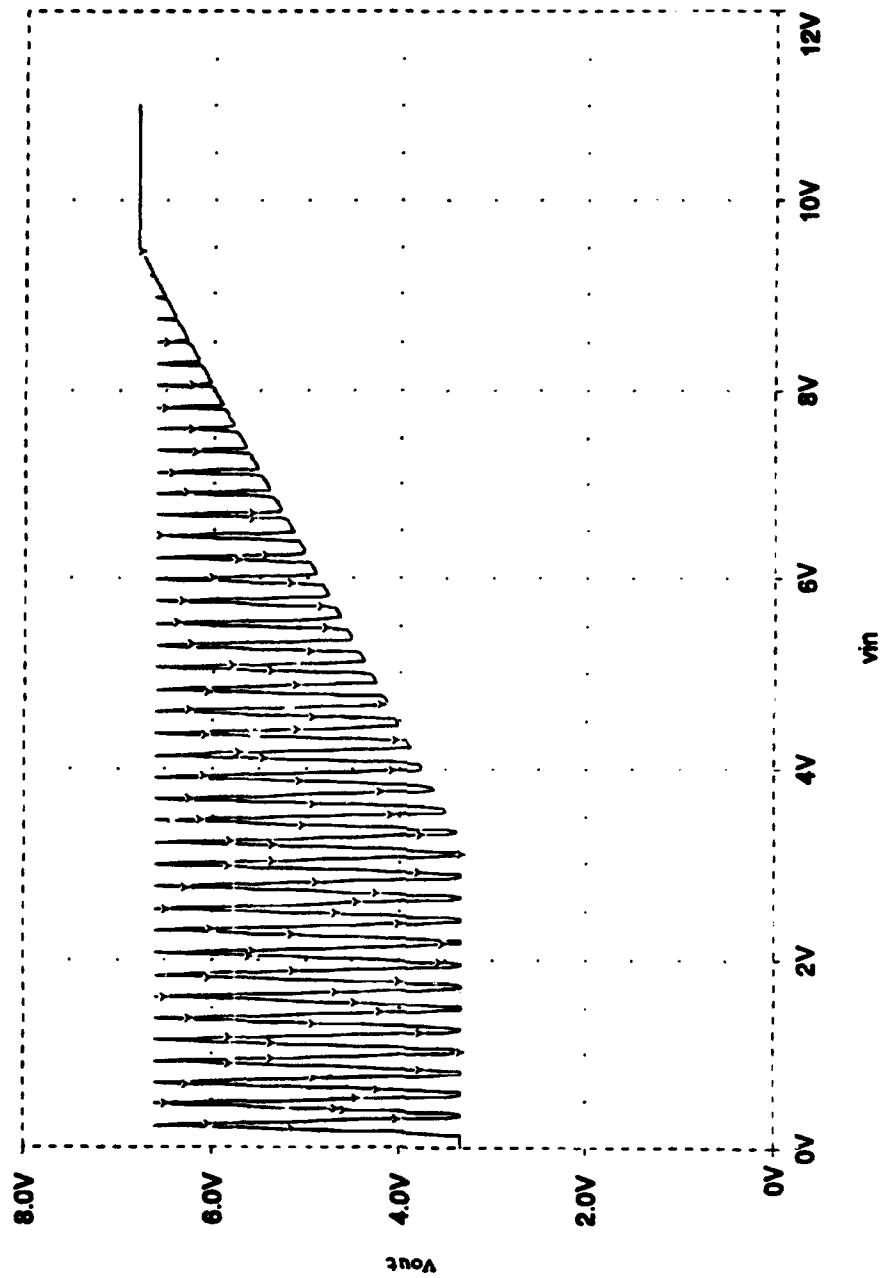


Figure 4.3: Folding circuit's output as the input approaches the value of the power supplies.

range for a particular modulus folding circuit must be adequately less than the value of the power supplies in order to avoid distortion. By supplying the proper equidistant reference voltages (separated by the modulus), the output folding waveform results as shown in Figure 4.4.

For the initial design, the folding circuits for a 10-bit SNS ADC were simulated using moduli $m_1=23$, $m_2=24$, $m_3=29$, and $m_4=31$. Using Equation 3.3 the SNS dynamic range was calculated to be

$$\frac{1}{2}m_2m_4 + m_1m_3 = 12(31) + 23(29) = 1039. \quad (4.1)$$

This SNS dynamic range was converted to an analog dynamic range of 5.2 volts by dividing by 200. Similarly, each modulus was divided by 200 to determine the reference voltages. This was done to enable the folding circuits to operate properly within the power supplies. Specifically, in the case of the modulus 23 folding circuit as shown in Figure 4.5 and 4.6, a linear ramp, V_{in} , covering the dynamic range was applied to the input. The folding waveform has a period of 0.115 volts ($\text{mod}23/200 = 0.115$). Each folded output will be quantized by $\lfloor m_i/2 \rfloor$ properly biased comparators. Each comparator must be biased to encode the input voltage correctly. In this case, the number of comparators, and

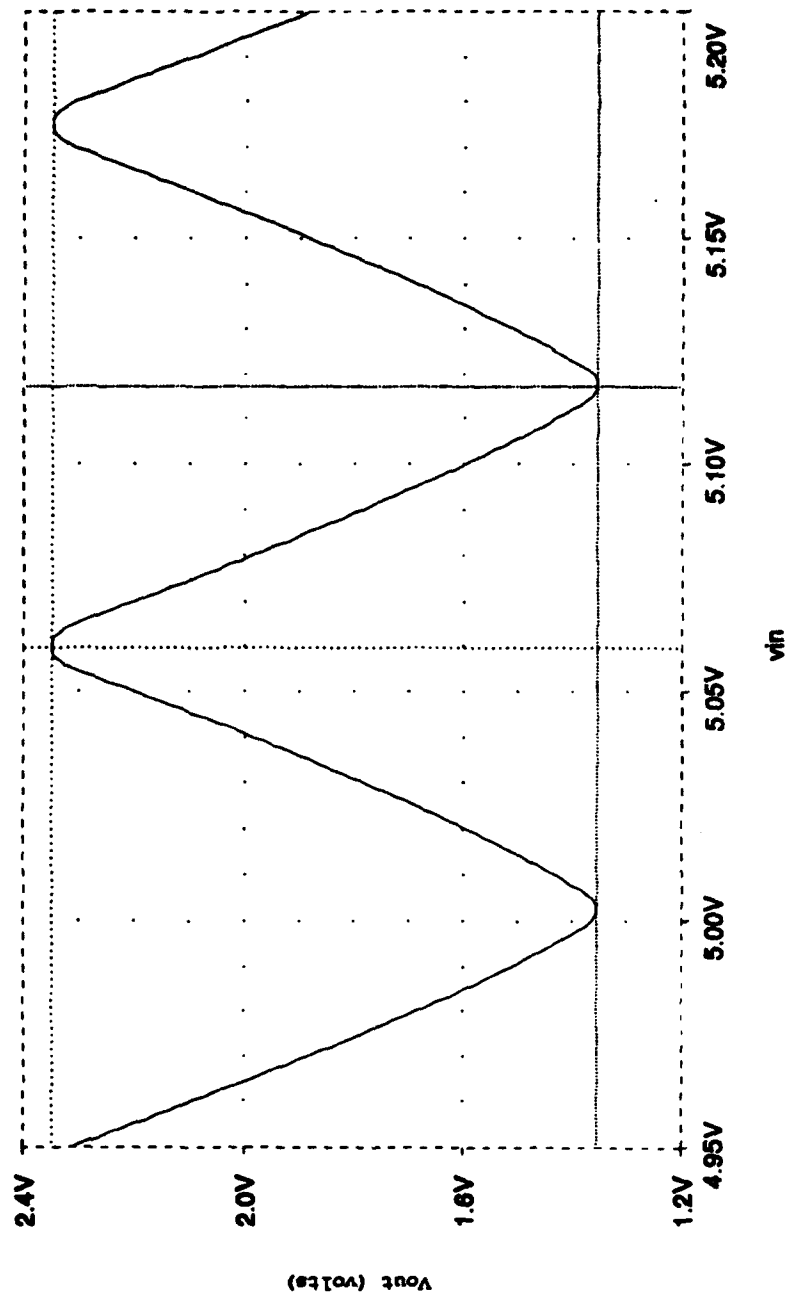


Figure 4.4: Periodicity in the folding circuit's output.

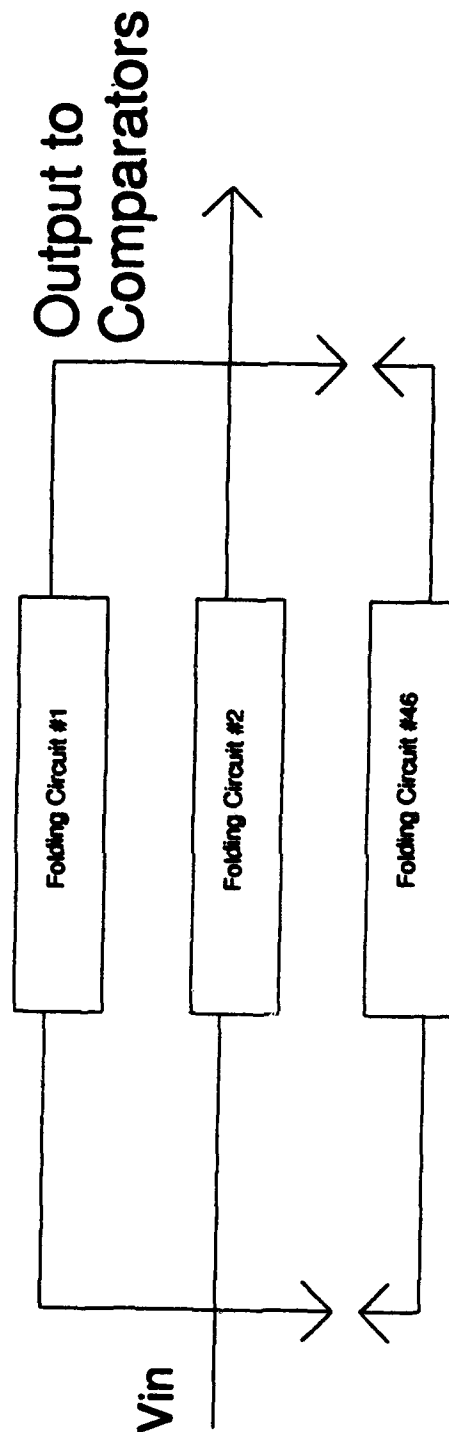


Figure 4.5: Modulus 23 folding circuit block diagram.

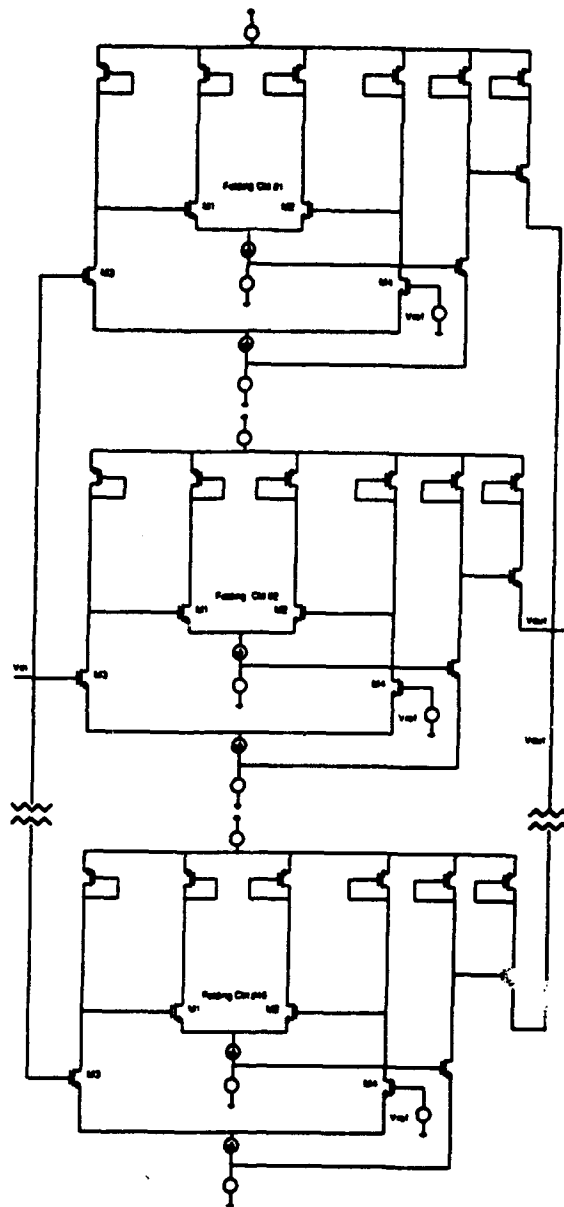


Figure 4.6: Modulus 23 folding circuit schematic.

consequently the number of threshold voltages, is equal to 11 ($\lfloor 23/2 \rfloor = \lfloor 11.5 \rfloor$). To arrive at the threshold voltages, the period of the fold is quantized by two times the desired number of threshold voltages. For a mod 23 folding circuit, this would mean that for every 5.23 mV change in V_m , starting from 0.0V up to the dynamic range, the value of the folded output would correspond to the threshold voltage level. The voltage threshold values for the mod 23 folding circuit are tabulated below in Table 4.1.

TABLE 4.1 VOLTAGE THRESHOLD LEVELS FOR A MOD 23 FOLDING CIRCUIT

Voltage Threshold Level	Value (Volts)
1	2.34
2	2.24
3	2.13
4	2.02
5	1.91
6	1.81
7	1.71
8	1.61
9	1.52
10	1.44
11	1.37

The results of the simulation of the mod23 folding circuit and connected comparators are shown in Figure 4.4 and Figure 4.7. As can be seen from the latter, the output of the

comparators are high as the output of the modulus folding circuit passes the determined comparator threshold voltages.

After the DC analysis was conducted, a transient analysis was performed on the mod23 folding circuit and connected comparators. This was done by ramping the input V_{in} through the dynamic range over a specific period of time. As can be seen in Figures 4.8-4.10, as the frequency of the ramped input increases, the output of the modulus folding circuit becomes distorted. To correct this problem, two possible solutions were explored. The first remedy was to reduce the width of the transistors comprising the differential amplifiers in the folding circuit. This would decrease the capacitance associated with the transistors and increase the gain. The second idea was to increase the amount of current flowing through current sources of the folding circuit.

Changing the geometries of the transistors had minimal effect on the transient response of the folding circuits. However, the opposite was true when the current of the folding circuits were increased. In Figures 4.11 and 4.12, the current sources of the folding circuits were 0.05 mA and 0.2 mA, respectfully. As can be seen from these simulation results, the increase in the current improved the frequency response from 100Hz to 100KHz. The frequency response can be further improved, but there is a drawback. As the current

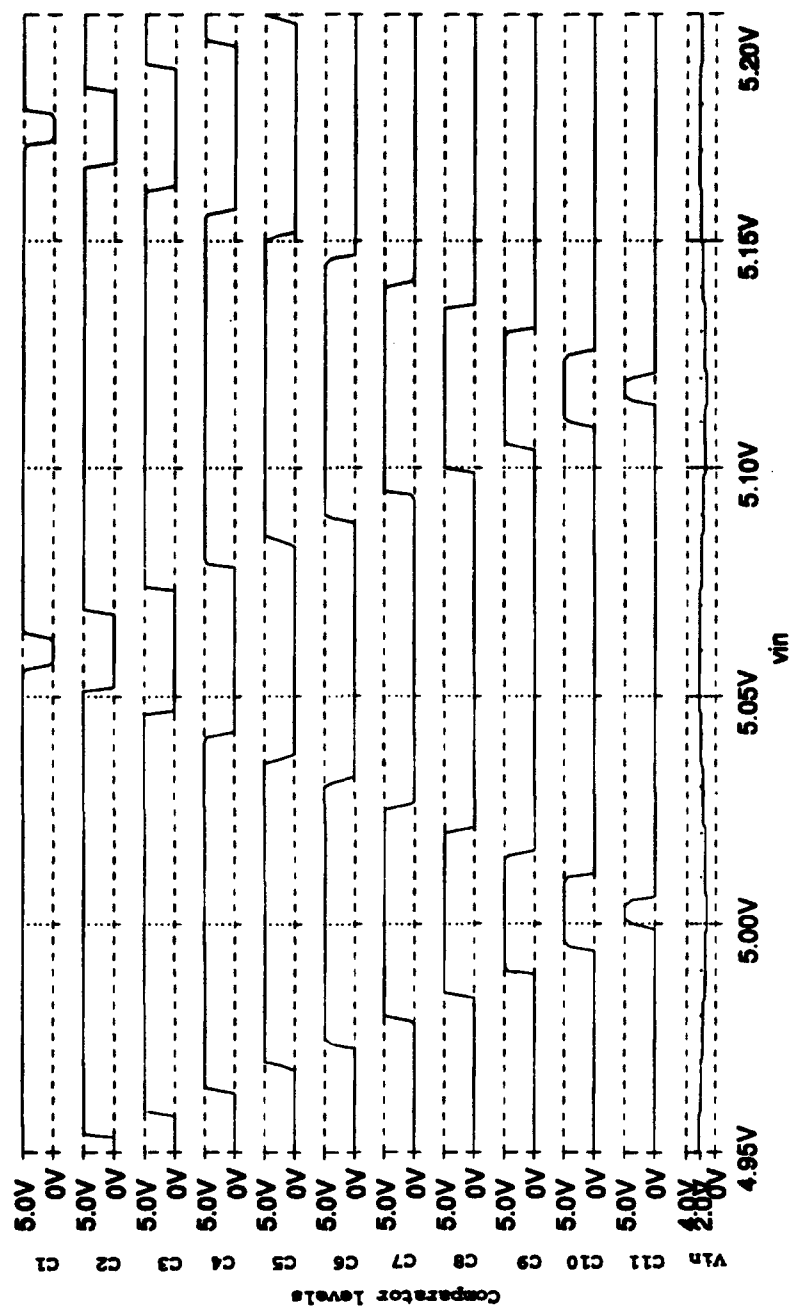


Figure 4.7: Comparator outputs of simulated mod 23 folding circuit.

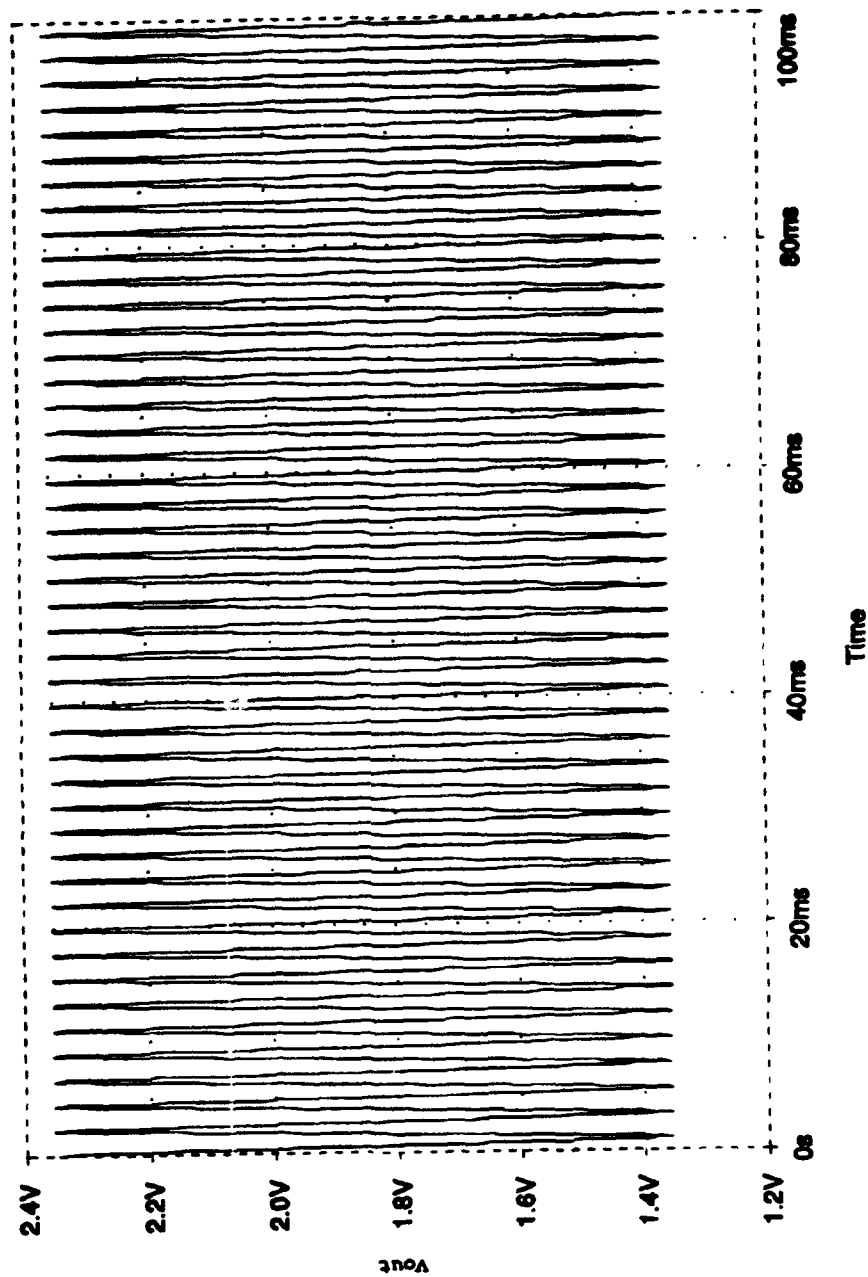


Figure 4.8: Mod 23 folding circuit output from a transient analysis at 10 Hz.

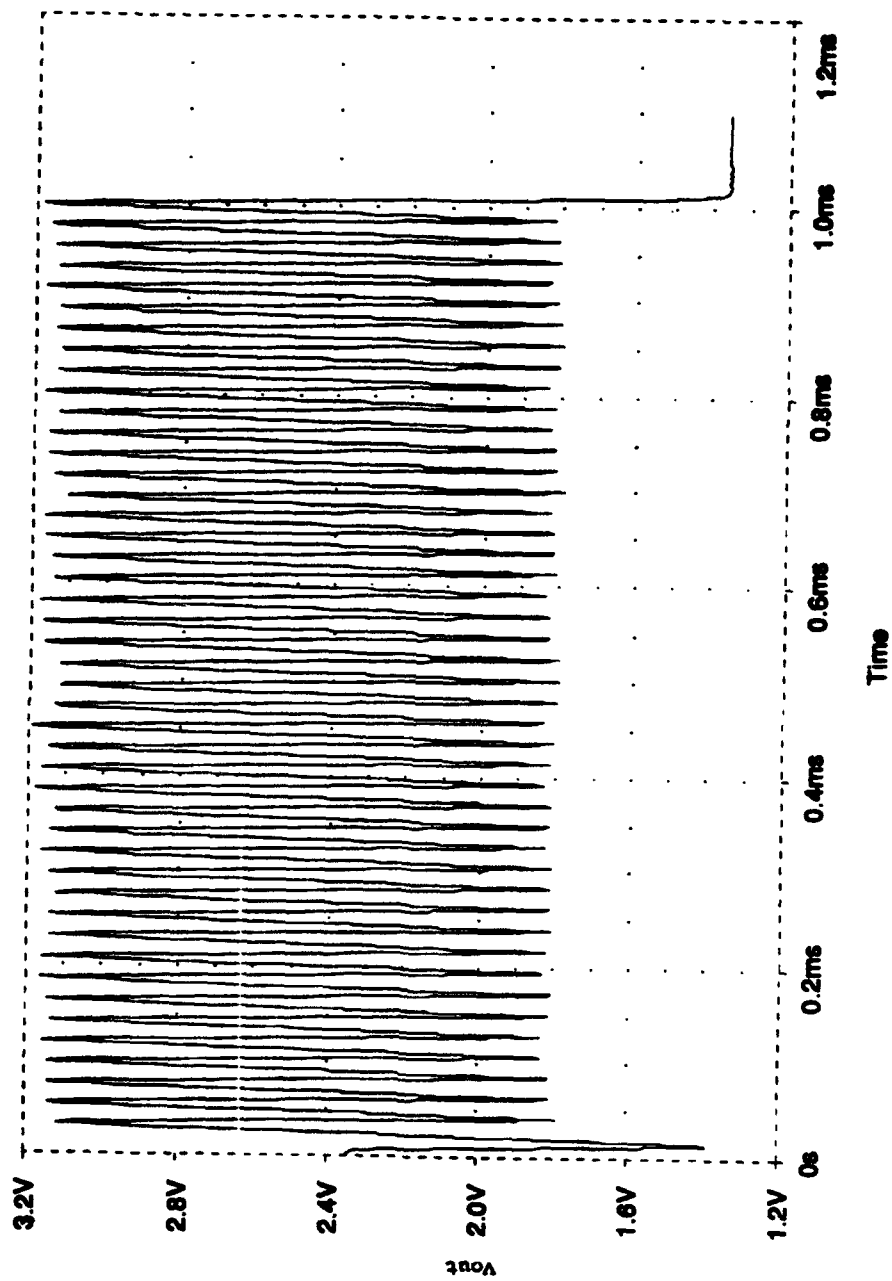


Figure 4.9: Mod 23 folding circuit output from a transient analysis at 1 KHz.

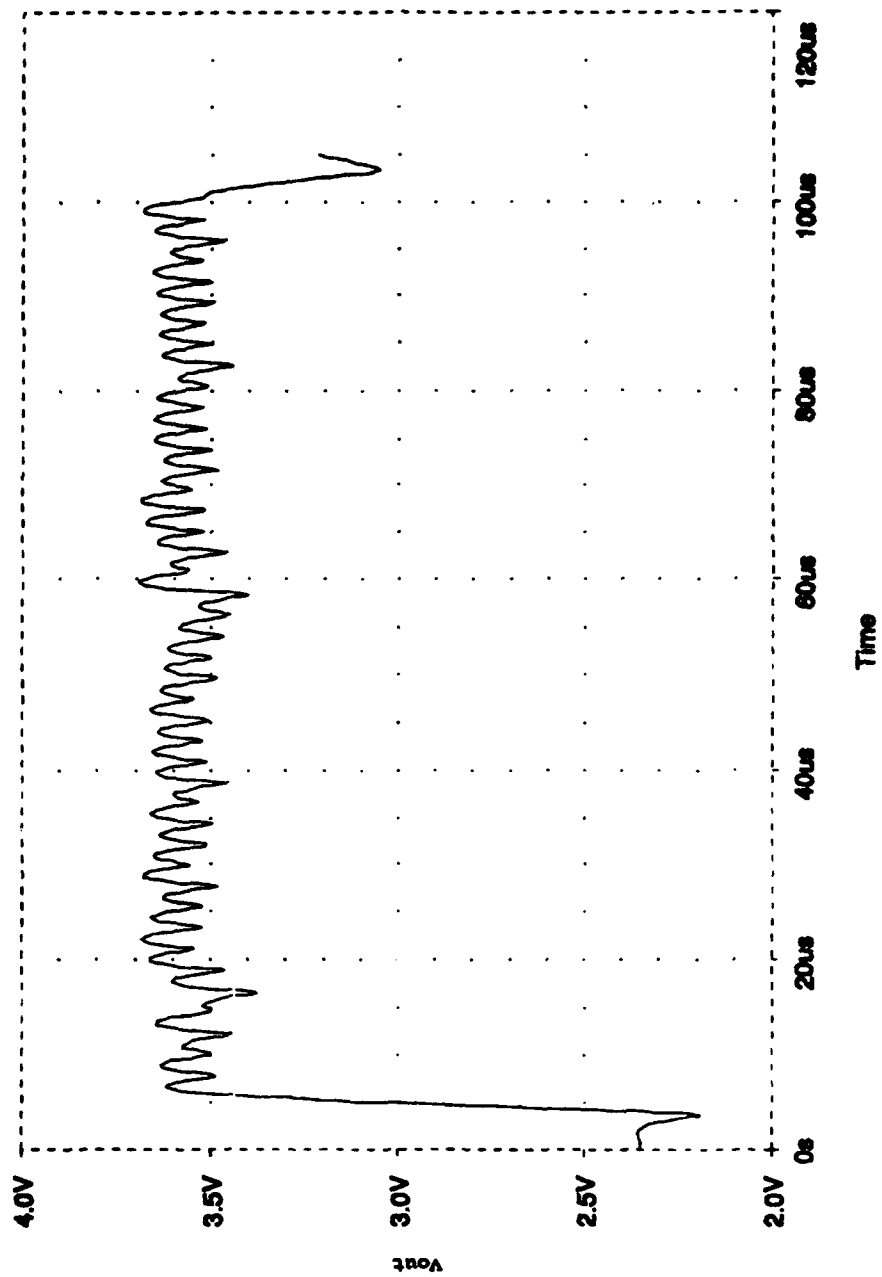


Figure 4.10: Mod 23 folding circuit transient analysis at 10 KHz.

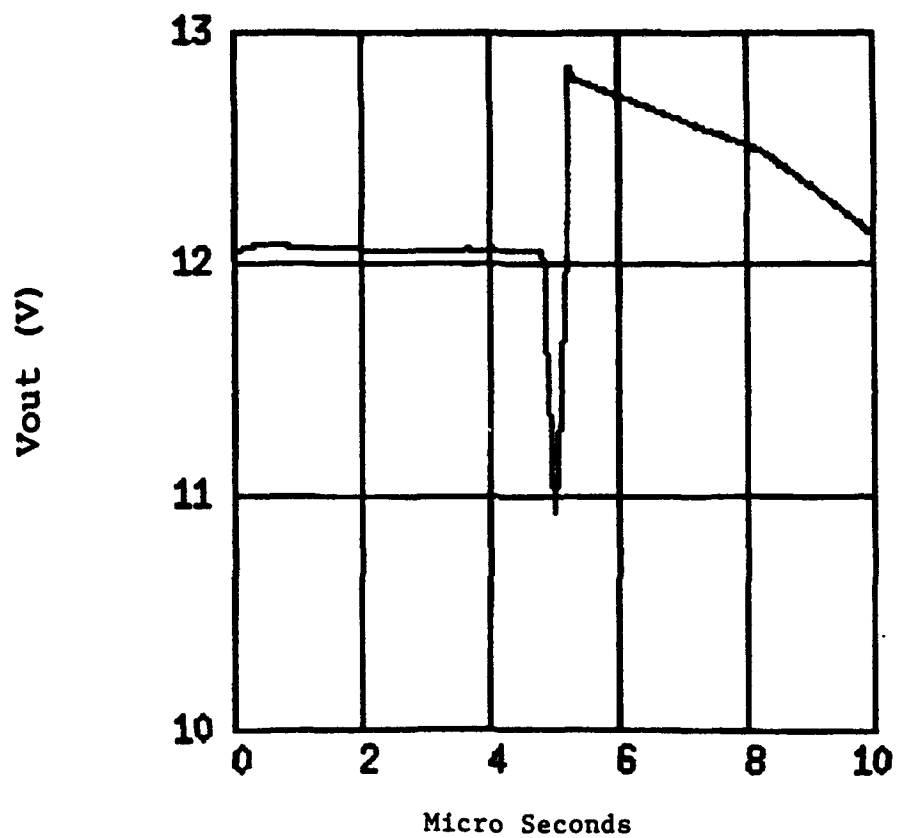


Figure 4.11: DC transfer curve of one fold with a 0.05 mA current source.

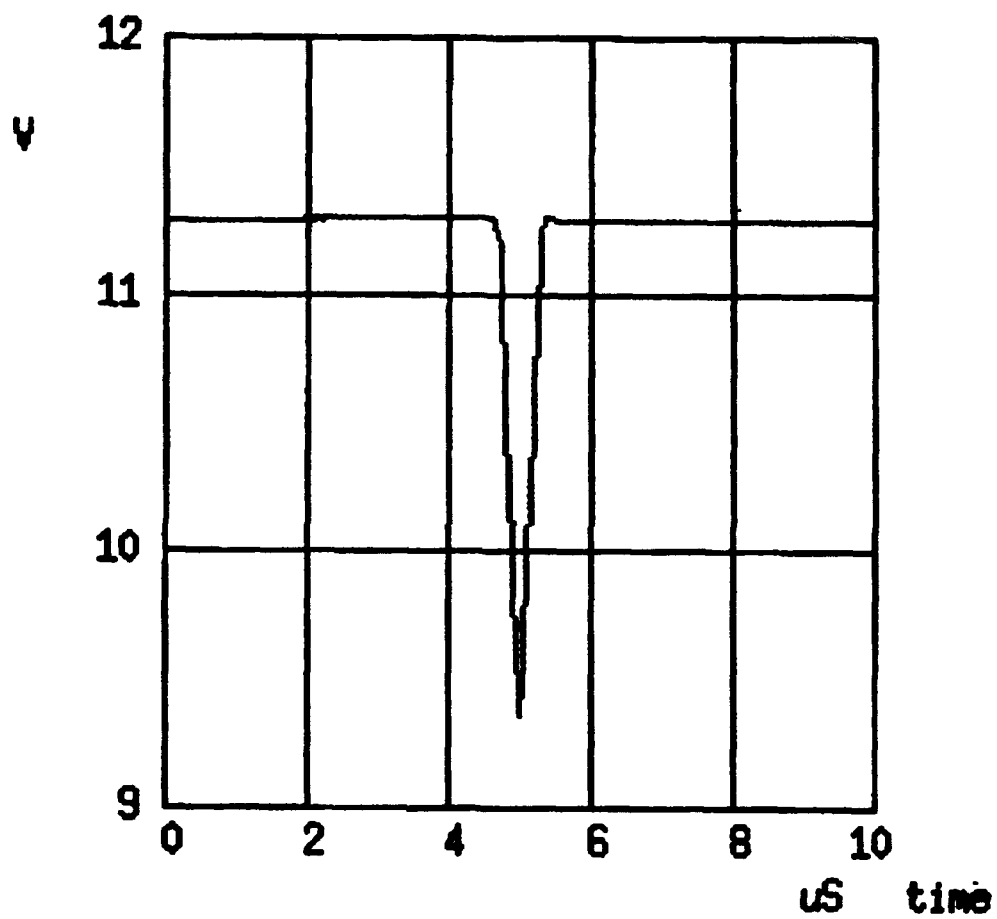


Figure 4.12: DC transfer curve of one fold with a current source of 0.2 mA.

flowing through the pair of differential amplifiers comprising the folding circuits was increased, the width of the folded output waveform expanded (see Figures 4.13 and 4.14). If the width of the folds become too large, then the required periodic folded output waveform cannot be achieved. Additionally, the power limitation of the device being fabricated is, in effect, an upper limit on how much the current sources can be increased. Thus, the frequency response that can be achieved in this particular design is dependent on the power limitation and on the semiconductor process that will be used for fabrication.

While investigating the frequency response difficulty, a new problem arose. As each folding circuit's output was connected together in parallel, the height of each of the folded output of the combined circuit decreased, as shown in Figure 4.15. One approach to solve this loading problem was to reduce the output resistance of the folding circuit. Various output amplifiers were connected to the folding circuits to try to reduce this value, but none solved the loading problem. One amplifier, called a voltage shifter and shown in Figure 4.16, was retained in the final design. As shown in Figure 4.17, it provided some flexibility in shifting the folded output waveform. After some adjustments to the current sources in the folding circuits, it was determined not to effect the loading problem. It was also found that by

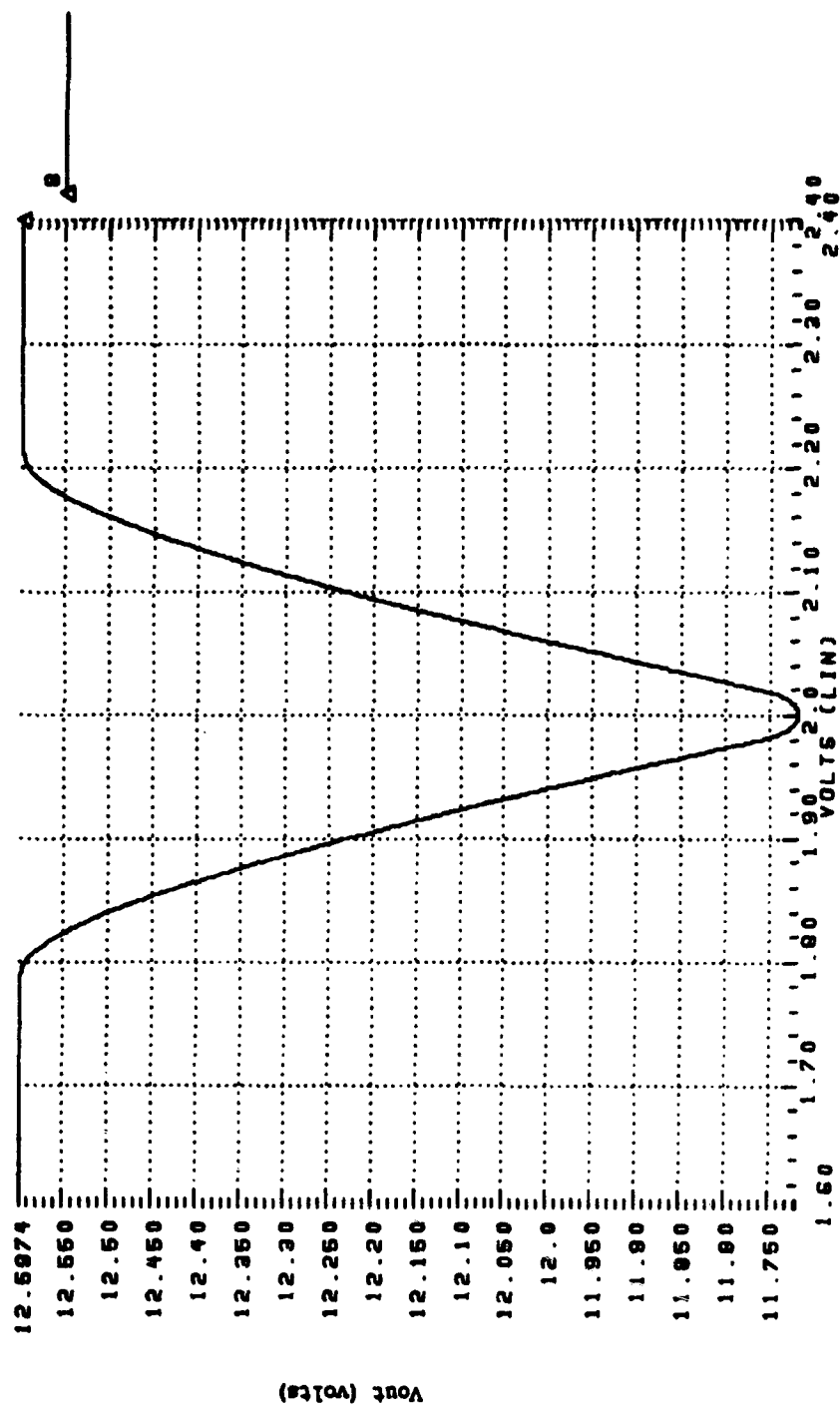


Figure 4.13: Narrow fold width due to small current source.

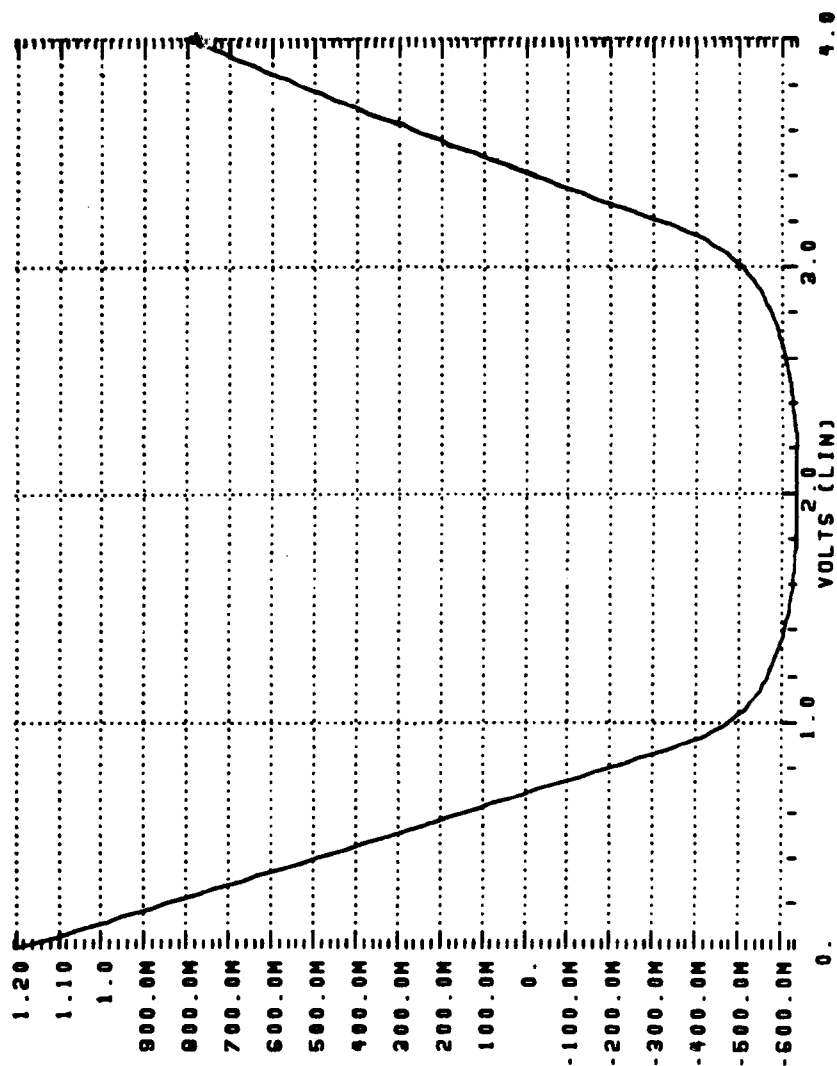


Figure 4.14: Wide fold width due to larger current source.

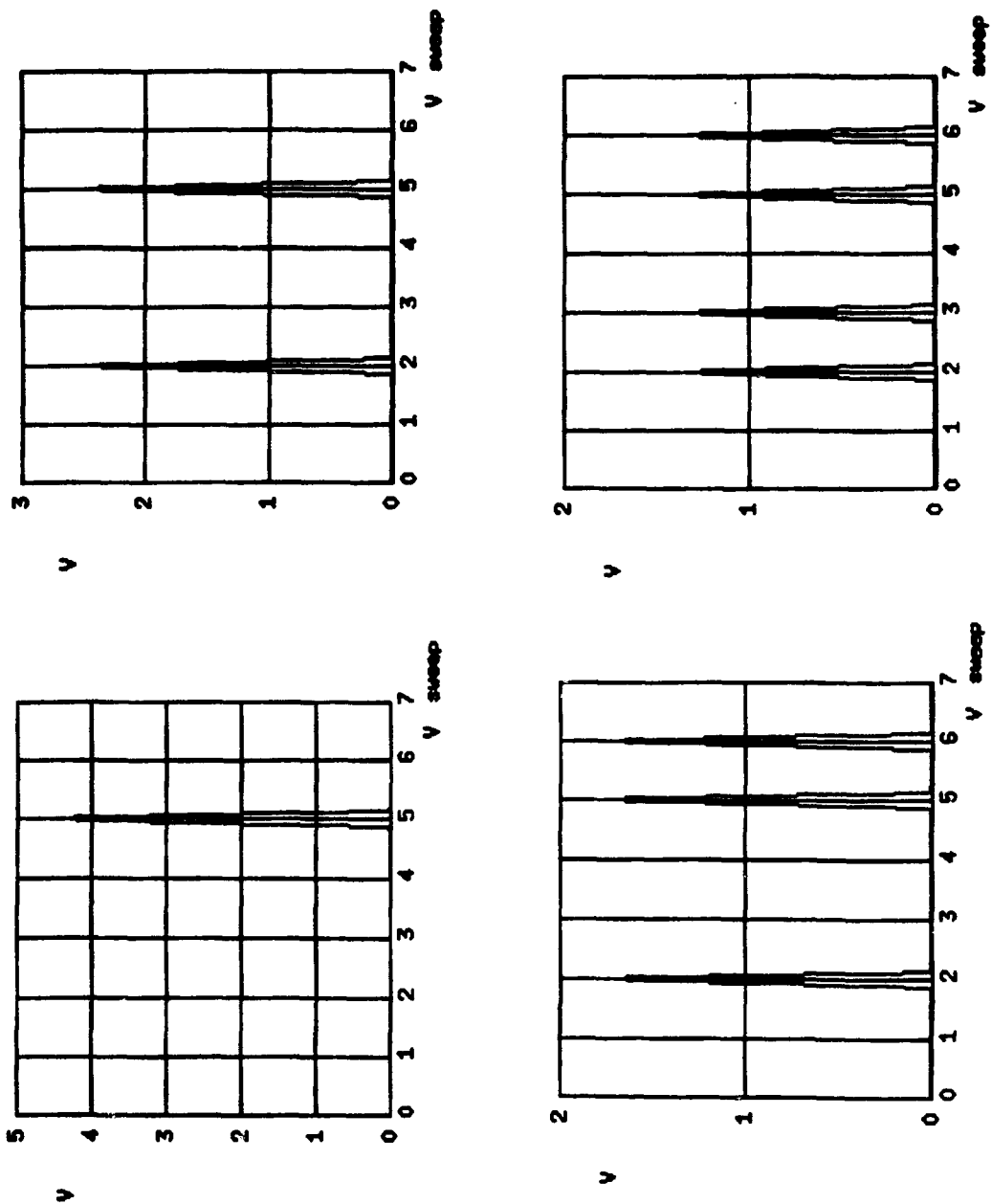


Figure 4.15: Loading problem in folding circuits.

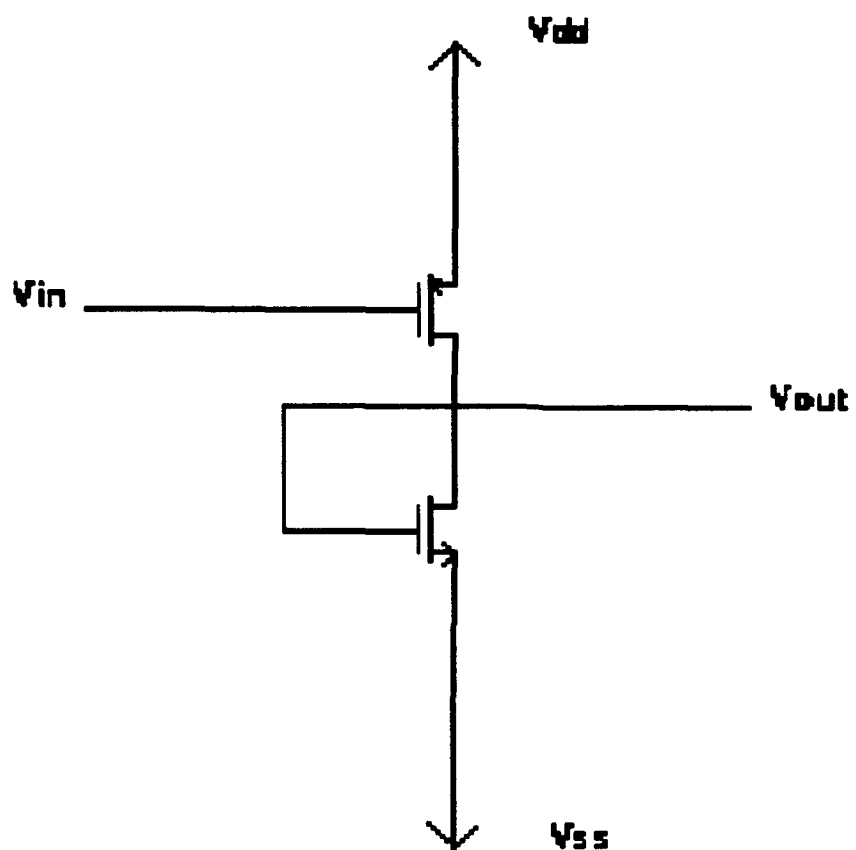


Figure 4.16: Voltage shifter circuit.

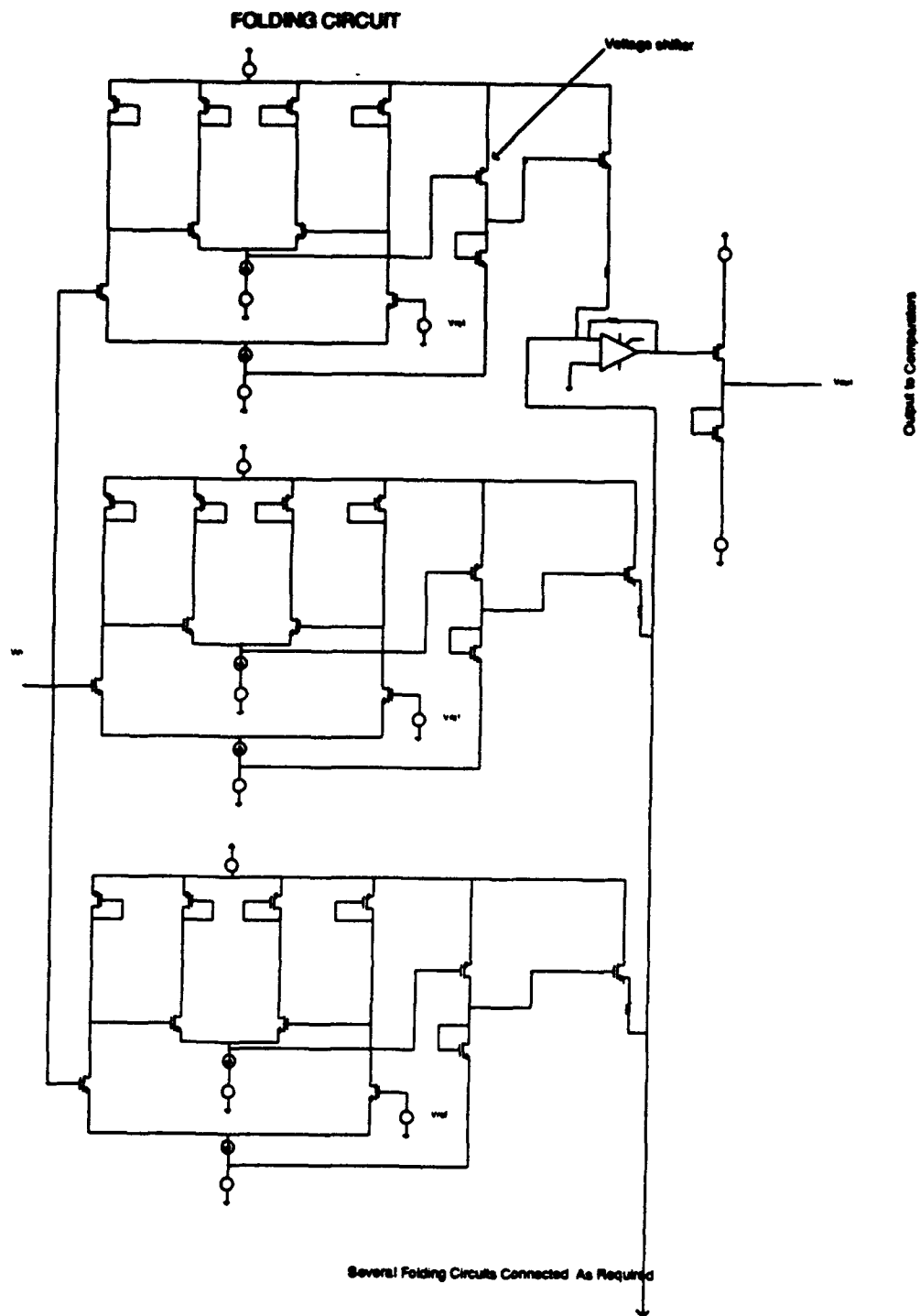


Figure 4.17: Revised modulus folding circuit.

removing the inverter portion of the inverter/source-follower amplifier of the folding circuit, the loading problem disappeared. The inverter was introducing some resistance as seen from the output. As a result, the output of the modulus folding circuit sees an increasing equivalent resistance as each folding circuit is connected in parallel.

Once the load problem was resolved, a modulus circuit was simulated by sweeping the input voltage from 0.0V to the dynamic range, which was 15.5V. As the simulation results in Figure 4.18 indicate, the height of the folds in the output are not consistent as the input voltage is swept. At first, it was believed that this fold height inconsistency was another loading problem and some investigation into the output resistance of the folding circuit was again undertaken. Specifically, the length and width of the MOSFET comprising the emitter follower was adjusted to achieve different gain values. However, this proved to be not productive. Another explanation for the fold height inconsistency was postulated to be the inability of the node, where all the folding circuits are connected together, to resolve its final value. Since all the folding circuit's outputs, with different output values at any given instant, are connected to this node, there might be a possibility that this node may not be able to settle to a final value. For this reason, an operational amplifier (op amp) in a summer configuration was introduced

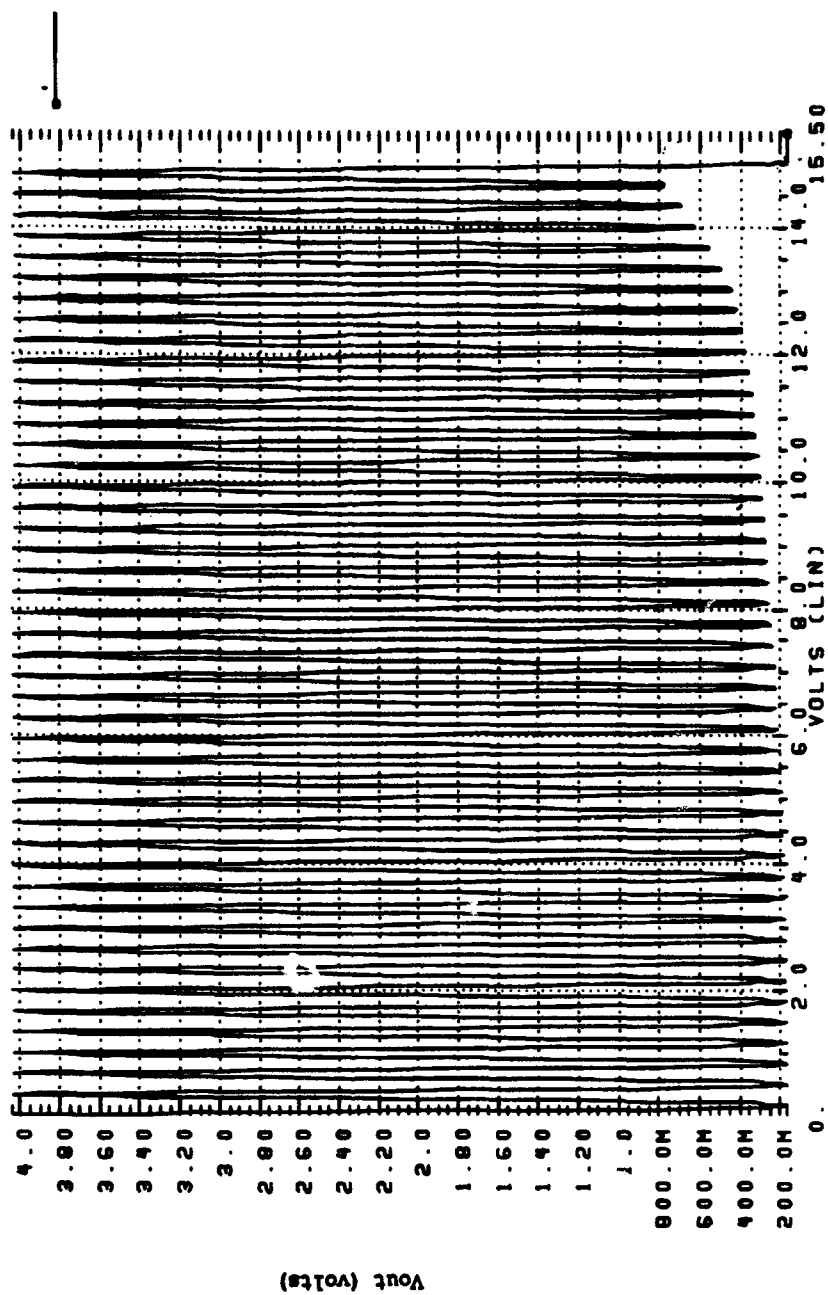


Figure 4.18: Fold height inconsistency in modulus folding circuit.

into the circuit, as shown in Figure 4.17. This summer would take the output of each folding circuit and sum it in a uniform manner, providing more stability for the output of the modulus folding circuit than without it. Yet, when this newly configured modulus folding circuit was simulated an unexpected result occurred. In Figures 4.19 and 4.20 the height of the folded output of the modulus folding circuit can be seen to increase and then to decrease as in the previous modulus folding circuits. Even though the opamp did not resolve this problem, it did produce an output that was shaped better than previous attempts and more conducive to the preprocessing operation.

Another test was devised to ascertain the reason for the fold height inconsistency problem. Instead of simulating the modulus folding circuit over the entire dynamic range, the simulation would be conducted in two parts, each part being associated with half the dynamic range. In each simulation only the affected folding circuits would actually be connected. If the beginning of the second simulation resulted in the output of the modulus folding circuit continuing at the point where the first half of the simulation ended, this would indicate that the problem lay somewhere in the individual folding circuit itself. If second-half simulation looked similar to the first-half simulation, then this would indicate that a loading problem exists. Figure 4.21 and 4.22 show the results of this test. As can be seen,

the test indicates that the problem is associated with the folding circuit itself.

To isolate the problem within the folding circuit, an intense investigation was conducted to determine the specific effect each element of the folding circuit had on the overall output. Table 4.2 tabulates the result of this investigation. After conducting this inquiry, it was established that the fold height inconsistency problem was a bias problem. This is because the reference voltages of each folding circuit increases in order to provide a folding process that accommodates the desired dynamic range. To compensate for the difference in the biasing of each folding circuit, the widths of the mosfets comprising the pair of diffamps were changed. Changing the widths of the mosfets kept the height and the width of each fold constant to produce a truly periodic folded output. This process took a considerable amount of time because each fold had to be simulated and adjusted individually.

Once the loading and fold height inconsistency problem was resolved, the design and simulation of the modulus folding circuits could go forward. However, some modification to the original design were made. In the final circuit, the comparator levels associated with each folding circuit were quite close. If the number of bits and numerical values of the moduli were lowered, then the total number of comparators and thus comparator levels would be smaller, as governed by

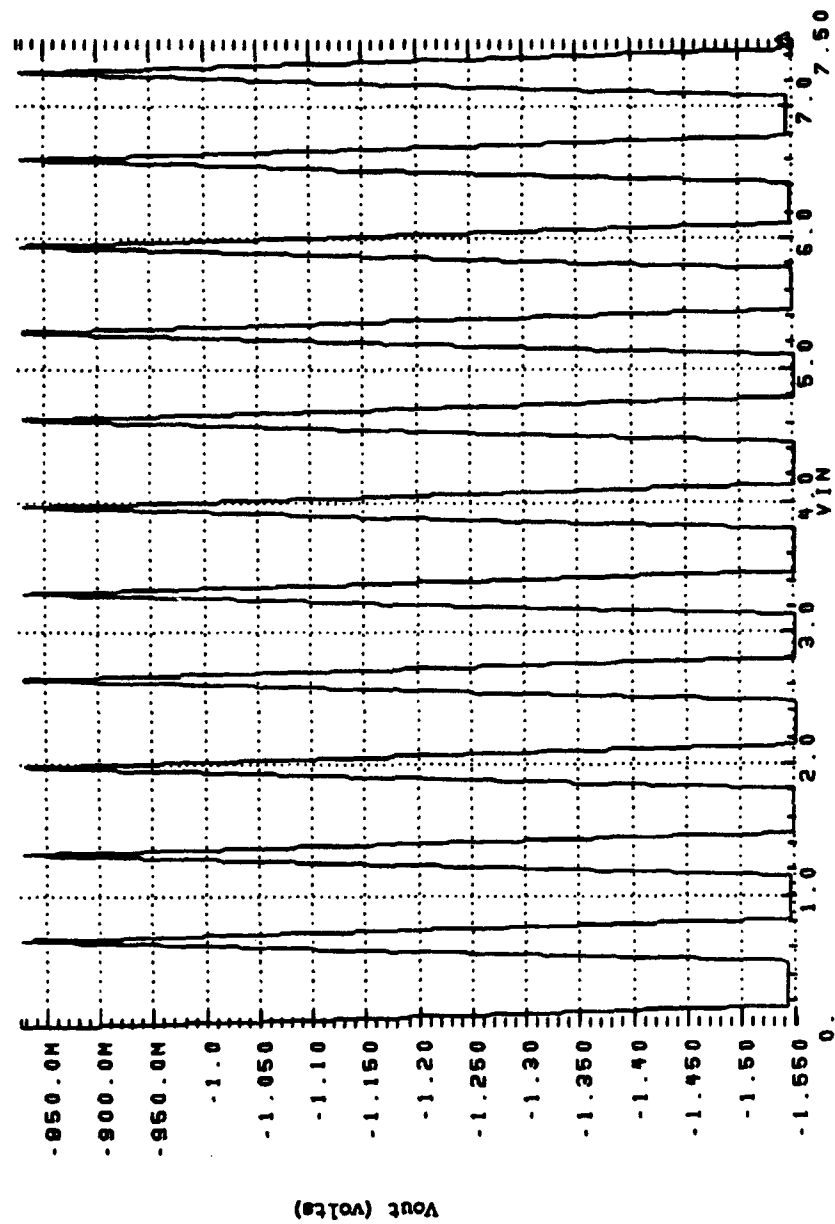


Figure 4.19: Fold height inconsistency variations.

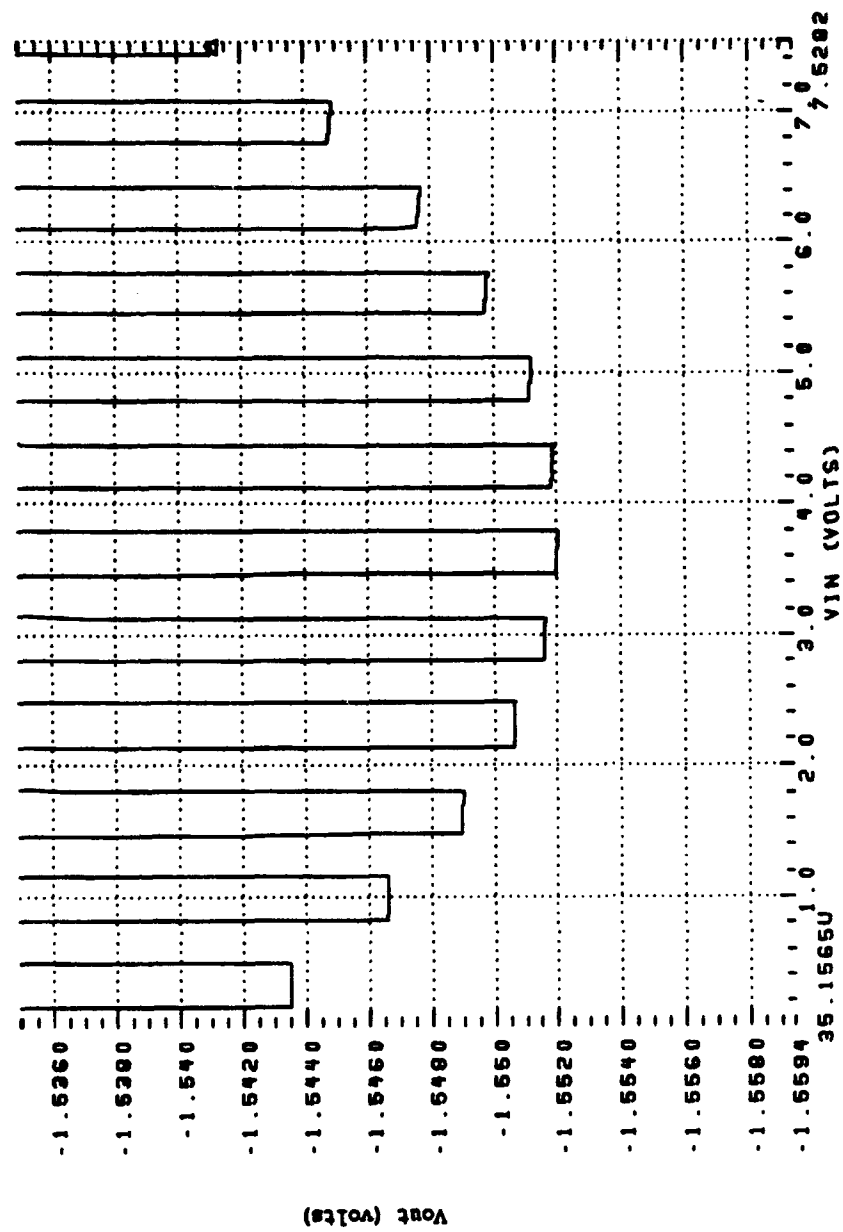


Figure 4.20: Bottom of folds from Figure 4.19.

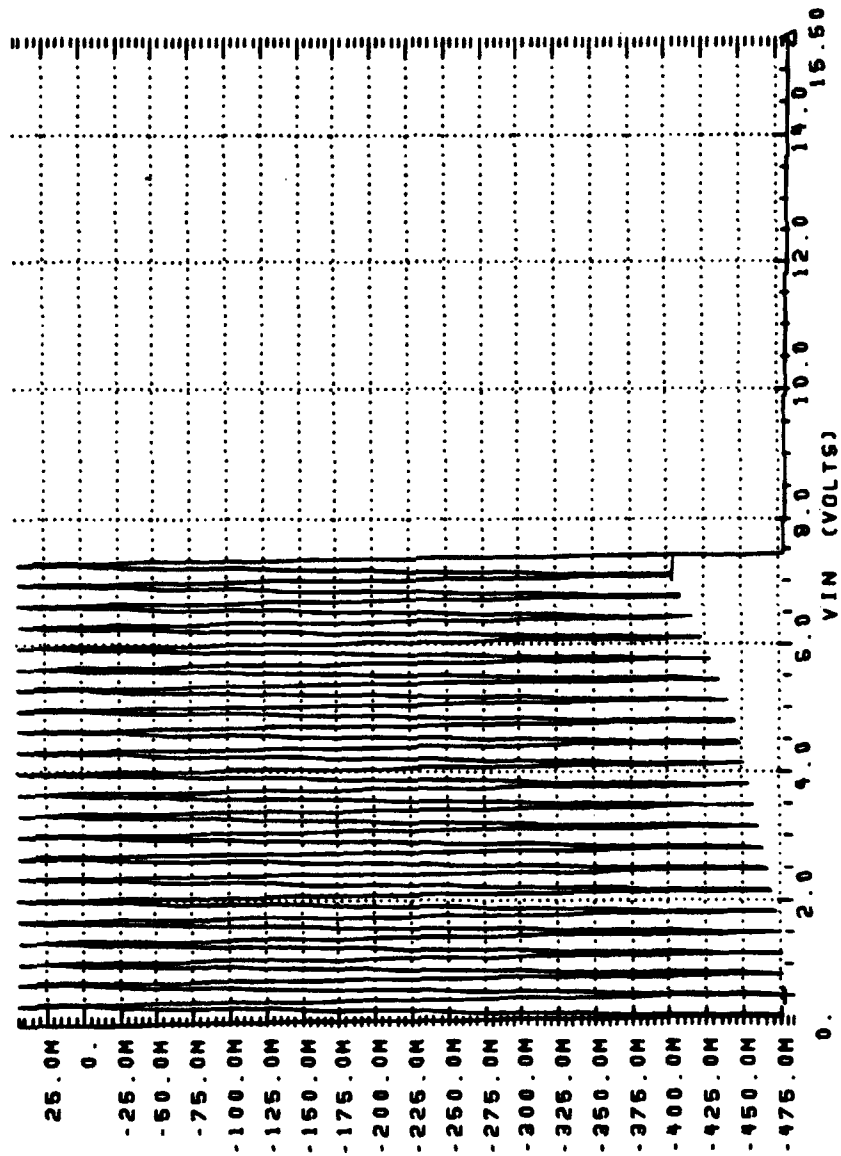


Figure 4.21: Fold height inconsistency test part 1.

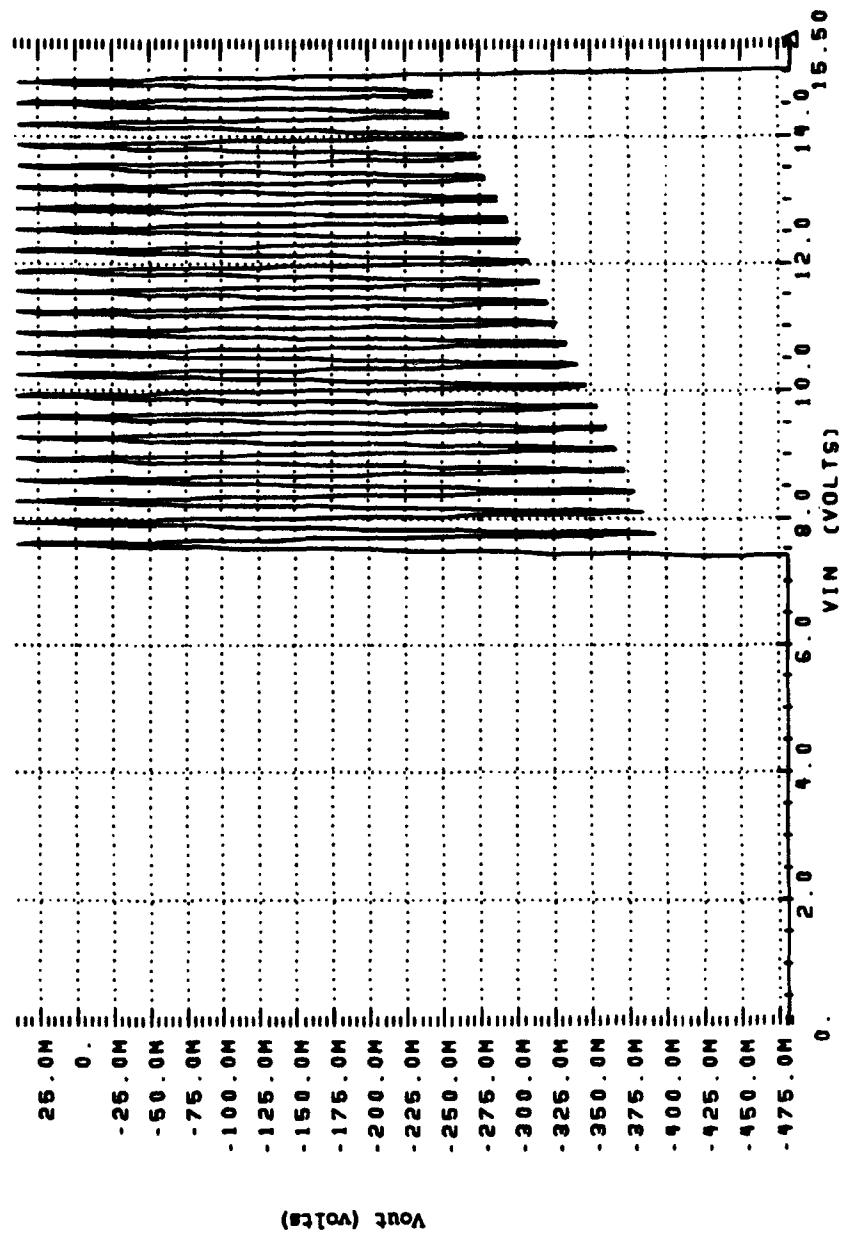


Figure 4.22: Fold height inconsistency test part 2.

TABLE 4.2 DEVICE PARAMETERS EFFECTS ON FOLDED OUTPUT

Elements	Parameter Changed	Effect on Fold
Internal Active Loads	Increase Width	Decrease Height of Fold
Voltage Shifter P _{fet}	Increase Width	Fold Shifts Up in Voltage
Emitter Follower	Increase Width	Fold Shifts Up in Voltage
External Active Load	Increase Width	Fold Height Increases
Upper Diffamp	Increase Width	Fold Height Increases
Lower Diffamp	Increase Width	Fold Width Decreases
Current Sources	Increase Current	Fold Height Decreases and Fold Width Increases

the equation $\lfloor m/2 \rfloor$. By keeping the same fold height, the distance between each comparator level increases. This aids the comparators to discriminate more effectively. Another modification to the original design was to increase the voltage dynamic range to 15.0V. In the original design, the width of the mosfets M3 and M4 in Figure 4.1 were inordinately large in order to achieve the required fold width, V_{width} . The fold width can be calculated by using the equation,

$$V_{width} = \frac{Vm_i}{M} \quad (4.2)$$

where V is the dynamic range as expressed in volts and M is the decimal value of the dynamic range. Table 4.3 gives some results of simulation which show the relationship of the widths of the mosfets M3 and M4 and the fold width.

TABLE
RELATIONSHIP BETWEEN DEVICE WIDTH
AND FOLD WIDTH

Width of M3 and M4 (μm)	Fold Width (volts)
400	0.37
300	0.42
200	0.51

For practical purposes, mosfet widths should be below 200 μm . From the above information and Equation 4.2, a determination can be made as to the minimum value of a modulus. For

instance, for a 10-bit resolution using 15.0V as a dynamic range, the minimum value for a set of moduli is

$$m_i = \frac{V_{width} M}{V} = \frac{(0.51)(1024)}{15} = 34. \quad (4.3)$$

However, for a 9-bit resolution using 15.0V as a dynamic range, the minimum value for a set of moduli is

$$m_i = \frac{(0.51)(512)}{15} \approx 17. \quad (4.4)$$

As can be seen, the change in using a 9-bit resolution and a 15.0V dynamic range is much more practical both in terms of mosfet size and the number of comparators.

To achieve a 9-bit resolution under the above constraints, the moduli chosen were 20,23,25,and 27. Using Equation 3.3, the SNS dynamic range for this set of moduli is

$$\hat{M} = \frac{1}{2}(20)(27) + (23)(25) = 845 \quad (4.5)$$

This is well above the 9-bit resolution of 512. Using Equation 4.2, the fold widths or periods for each of the moduli are

$$V_{width} = \frac{(15V)(20)}{512} = 0.59V \quad (4.6)$$

$$V_{width_{23}} = \frac{(15V)(23)}{512} = 0.67V \quad (4.7)$$

$$V_{width_{25}} = \frac{(15V)(25)}{512} = 0.73V \quad (4.8)$$

$$V_{width_{27}} = \frac{(15V)(27)}{512} = 0.79V. \quad (4.9)$$

Figure 4.23 shows the output of the mod 20 folding circuit, demonstrating the validity of the derived design parameters.

Before further progress was made with this modified design, a new version of the symmetrical number system was introduced, as was discussed in Chapter III. Because of the benefits discussed in Chapter III, the decision was made to modify the design again in order to incorporate the new symmetrical number system as the basis for the analog preprocessing architecture. To achieve a 9-bit resolution, the set of moduli chosen were $m_1=7$, $m_2=8$, and $m_3=11$. Using Equation 3.6, the new SNS dynamic range for this set of moduli is

$$\hat{M} = \prod_{i=1}^n m_i = (7)(8)(11). \quad (4.10)$$

This is well above the 9-bit resolution of 512. Under this new symmetrical number system, the fold width of period can be calculated to be

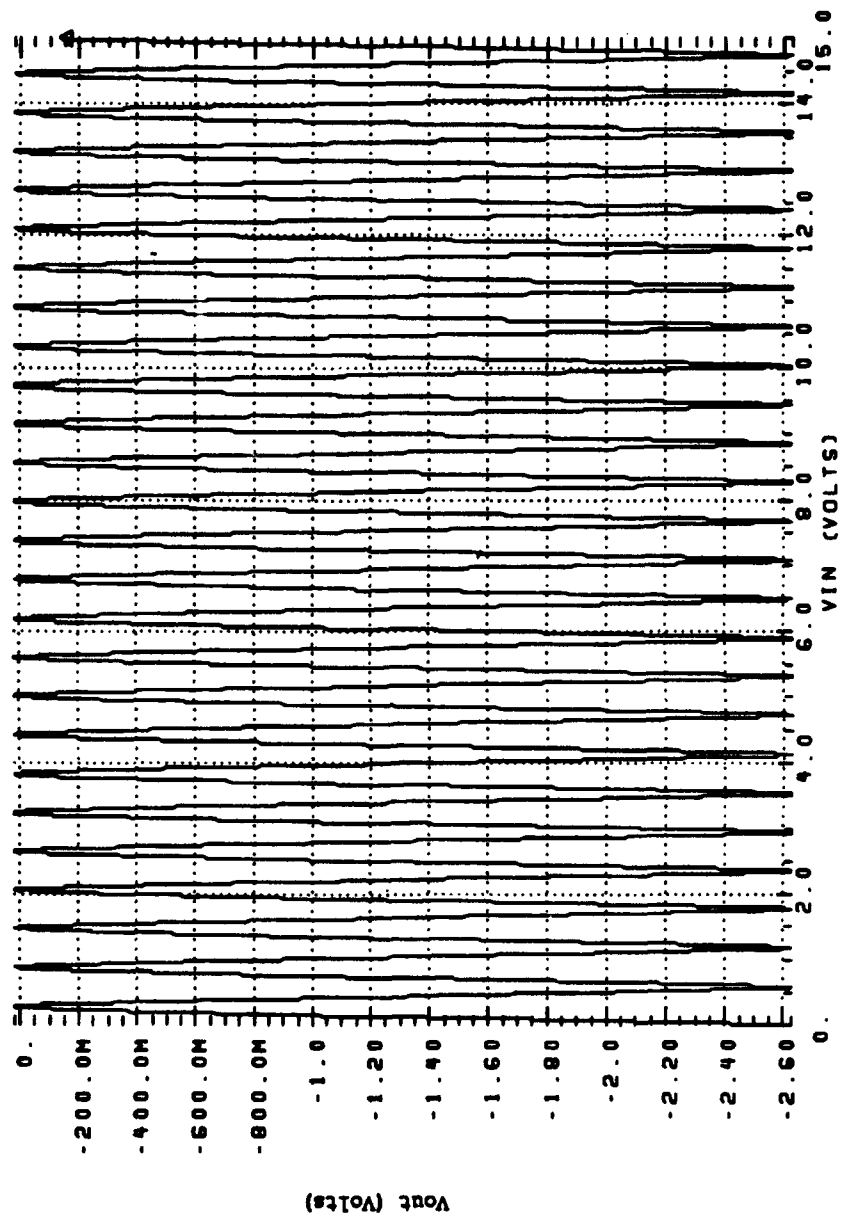


Figure 4.23: Modulus 20 folding circuit output.

$$V_{width} = \frac{2m_i V}{M}. \quad (4.11)$$

Using Equation 4.11, the fold width or period for each of the moduli is

$$V_{width_7} = \frac{(15V)(14)}{512} = 0.41V \quad (4.12)$$

$$V_{width_8} = \frac{(15V)(16)}{512} = 0.47V \quad (4.13)$$

$$V_{width_{11}} = \frac{(15V)(22)}{512} = 0.67V. \quad (4.14)$$

From these parameters and the generic modulus folding circuit in Figure 4.1, each modulus folding circuit as was designed and simulated by applying an input which ramped from 0.0V to 15.0V. Figures 4.24-26 show the results of these simulations which validate the strength of the design. Figures 4.27-29 were used to determine the comparator levels for each modulus folding circuit. In the new SNS, the number of comparator levels required is m_i-1 . As was discussed before in the beginning of this chapter, the comparator threshold levels are determined by quantizing the period of a particular fold by two times the desired number of threshold voltages. As the input voltage, V_m , arrives at one of these quantization points as it sweeps from 0.0V to 15.0V, the value of the folded output at that point would correspond to a threshold voltage level. The threshold voltages for each of the moduli used in this final design are tabulated in Tables 4.4-4.6.

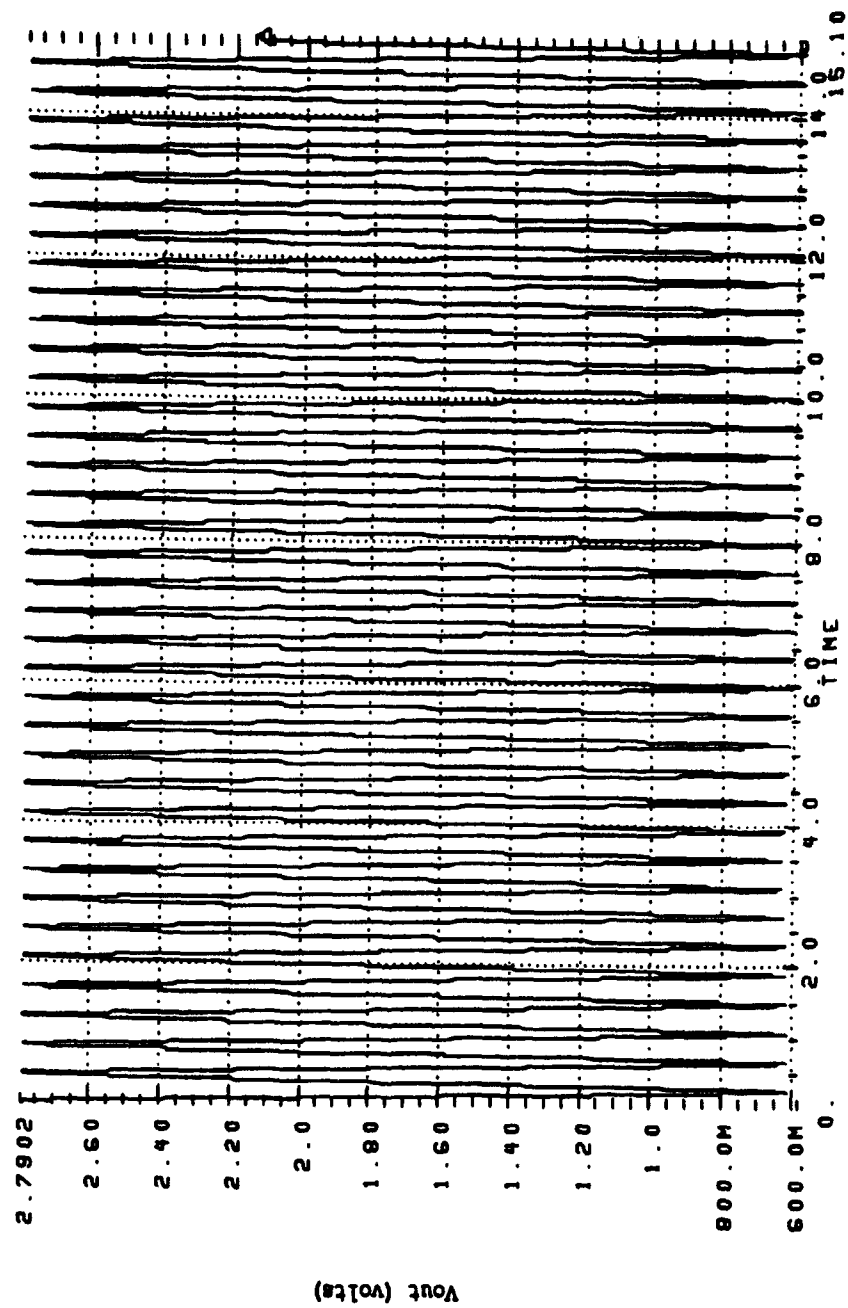


Figure 4.24: Modulus 7 folding circuit output.

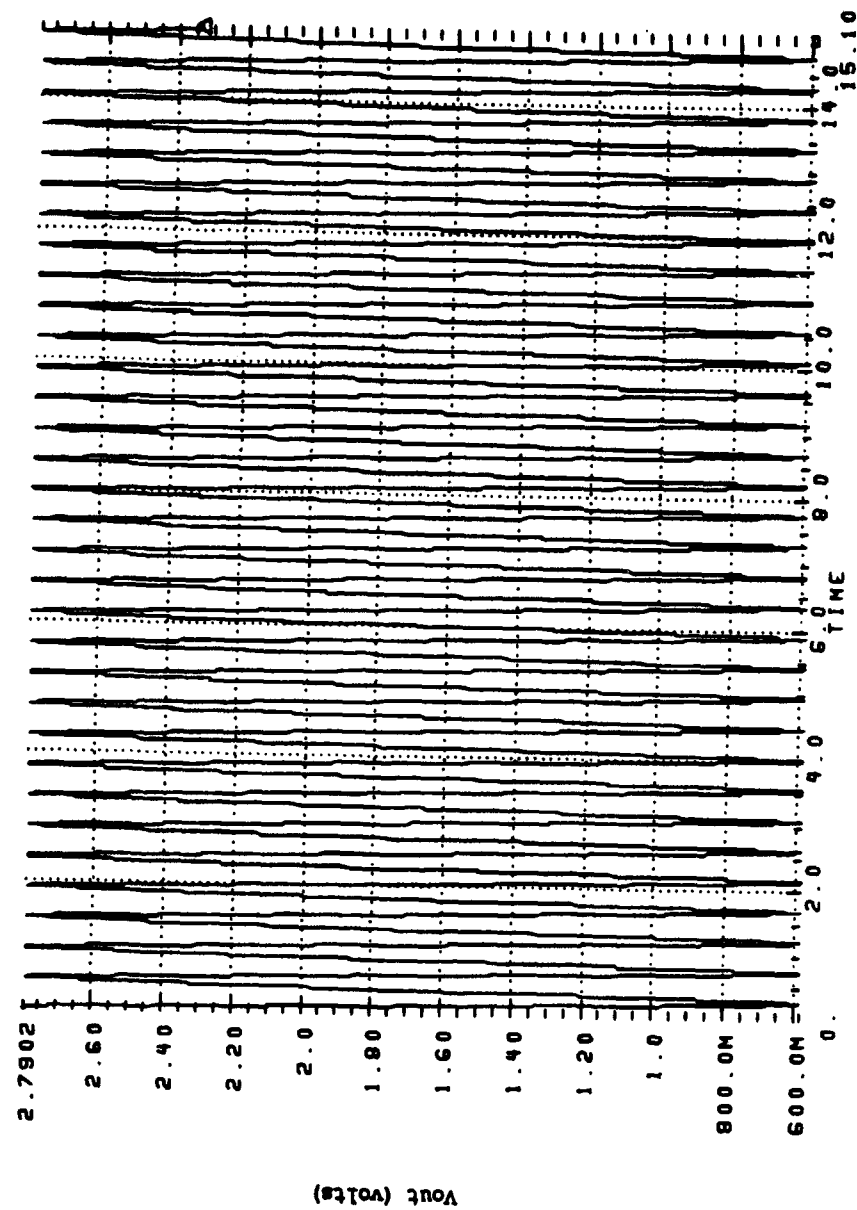


Figure 4.25: Modulus 8 folding circuit output.

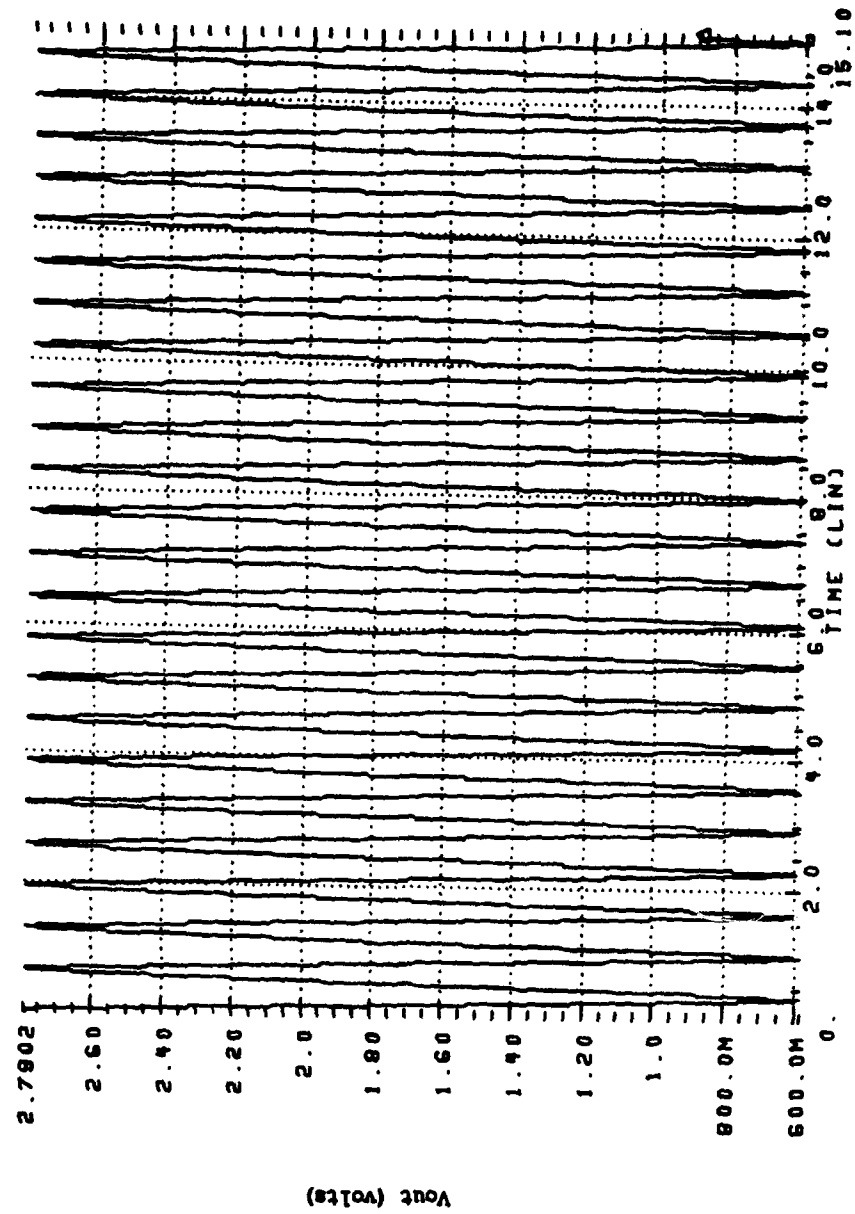


Figure 4.26: Modulus 11 folding circuits output.

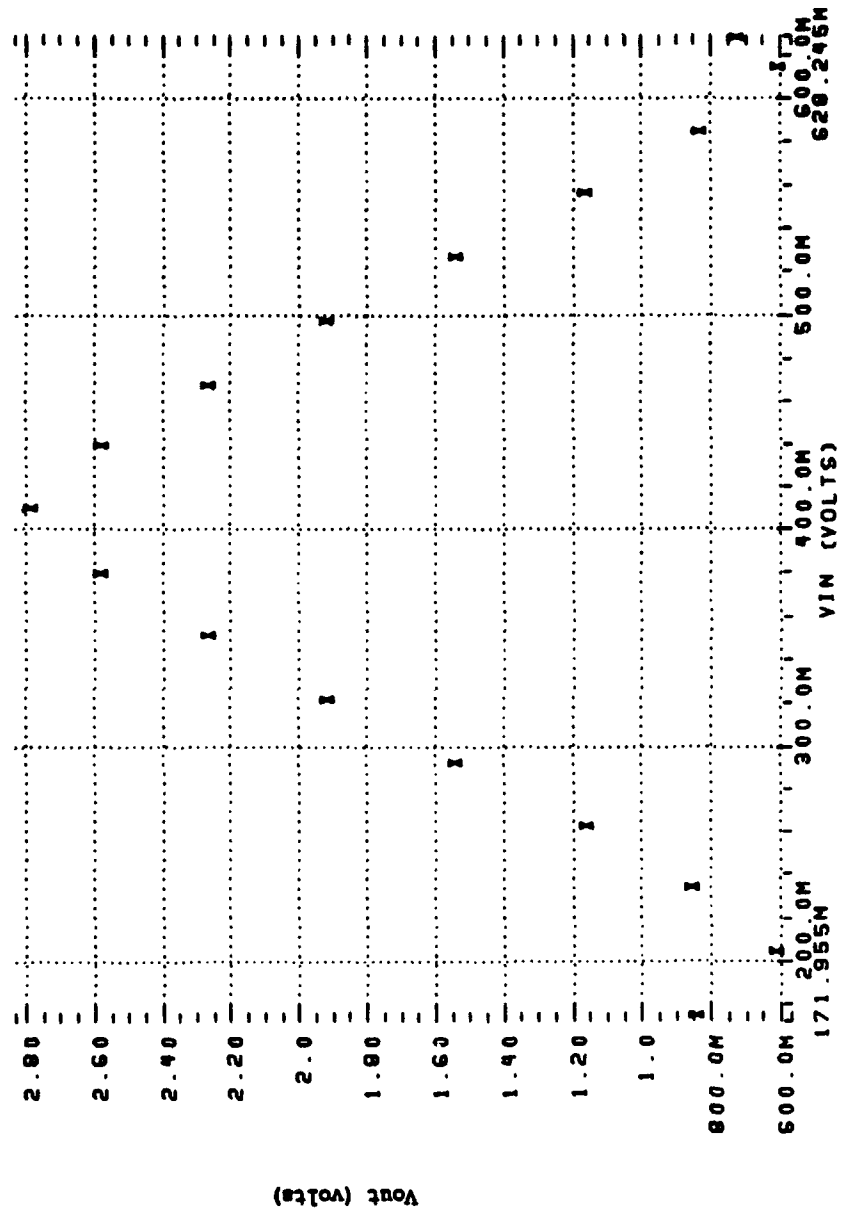


Figure 4.27: Modulus 7 comparator levels.

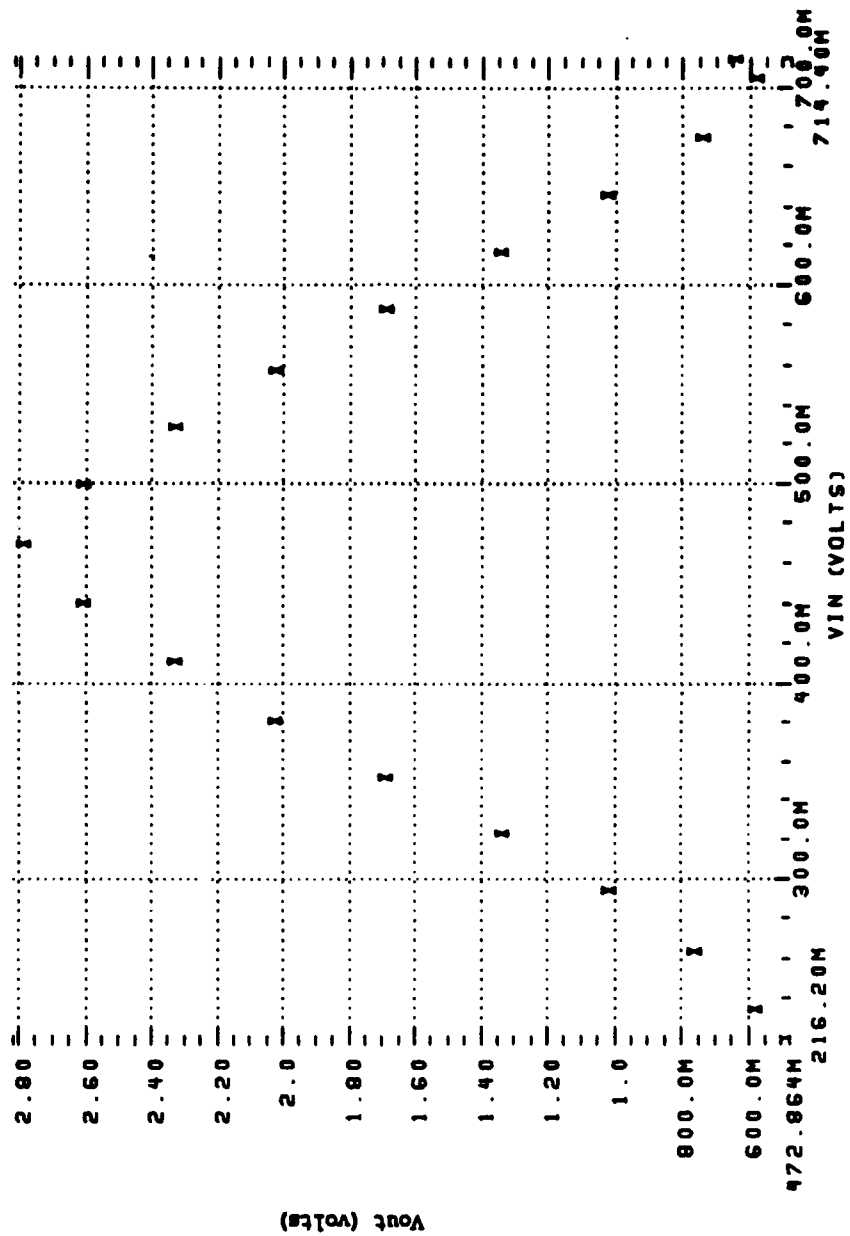


Figure 4.28: Modulus 8 comparator levels.

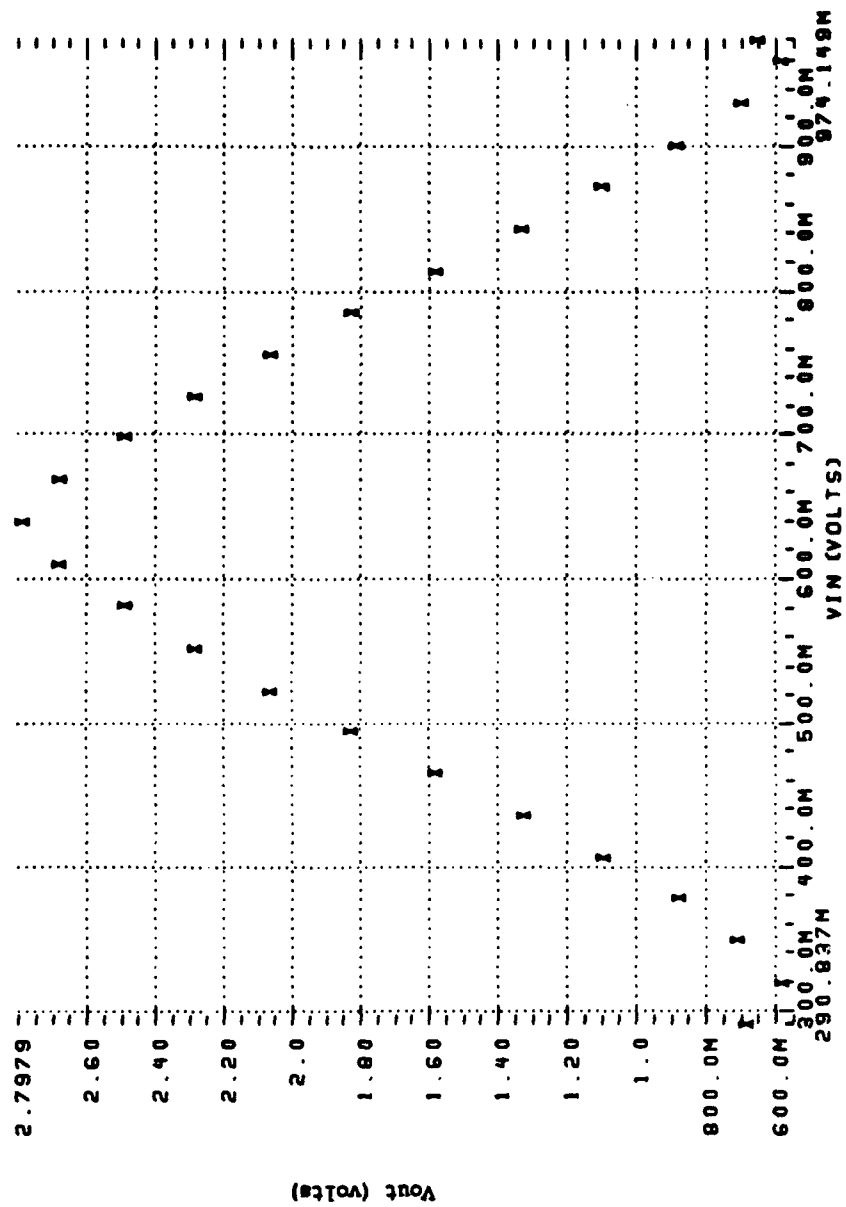


Figure 4.29: Modulus 11 comparator levels.

TABLE 4.4 MOD 7 COMPARATOR LEVELS

Voltage Threshold Level	Value (Volts)
1	2.58
2	2.27
3	1.92
4	1.54
5	1.16
6	0.858

TABLE 4.5 MOD 8 COMPARATOR LEVELS

Voltage Threshold Level	Value (Volts)
1	2.61
2	2.33
3	2.02
4	1.69
5	1.34
6	1.02
7	0.762

TABLE 4.6 MOD 11 COMPARATOR LEVELS

Voltage Threshold Level	Value (Volts)
1	2.68
2	2.49
3	2.28
4	1.07
5	1.83
6	1.58
7	1.33
8	1.09
9	0.882
10	0.714

2. Comparator Design and Simulation

After the design of the folding circuits were finalized, the next step was to design and simulate a comparator circuit which would not only effectively quantize the folded output but be able to effectively sample it. The sample-and-hold feature is important in order for the comparators to be able resolve a quickly changing circuit. The first attempted design was a clocked comparator shown in Figure 4.30. In this design, two non-overlapping clocks are applied to the gates of a pair of mosfets. One of these mosfets' drain is connected to output of the modulus folding circuit, while the other mosfets' drain is connected to a predetermined threshold voltage. The operation of this comparator is divided into two phases, charge and evaluate. During the charge phase, the CLK1 signal is high, and the CLK2 signal is low. At this point, both the input voltage from the output of the modulus folding circuit and the threshold voltage are sampled. During the evaluate phase, the CLK1 signal is low and the CLK2 signal is high. The diffamp comprised of mosfets M4 and M7 will recursively evaluate the values of the input and the threshold voltages. If the threshold voltage is larger than the input voltage, then the comparator output will be high, or true as shown in Figure 4.31. If the input voltage is higher than the threshold

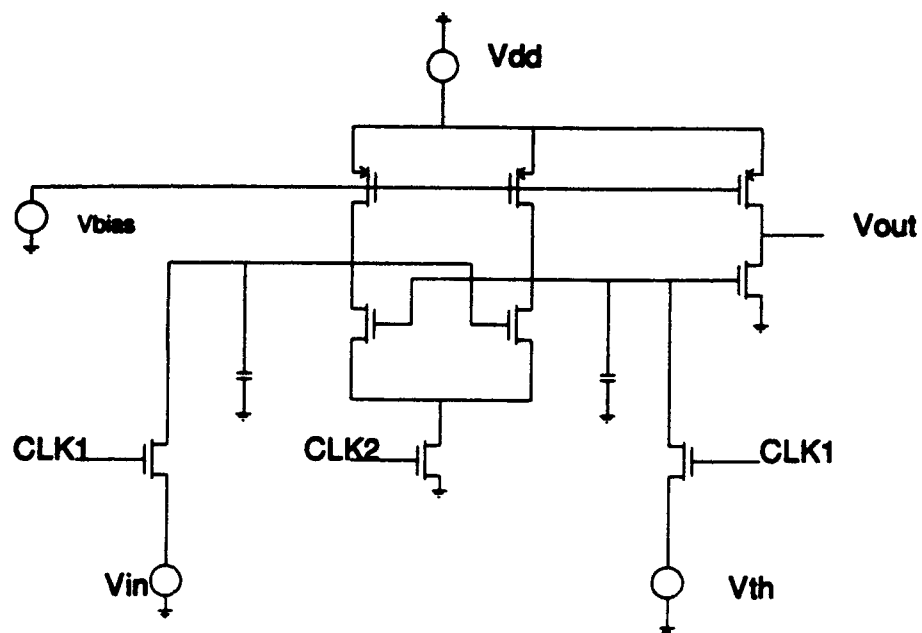


Figure 4.30: Clocked comparator circuit diagram.

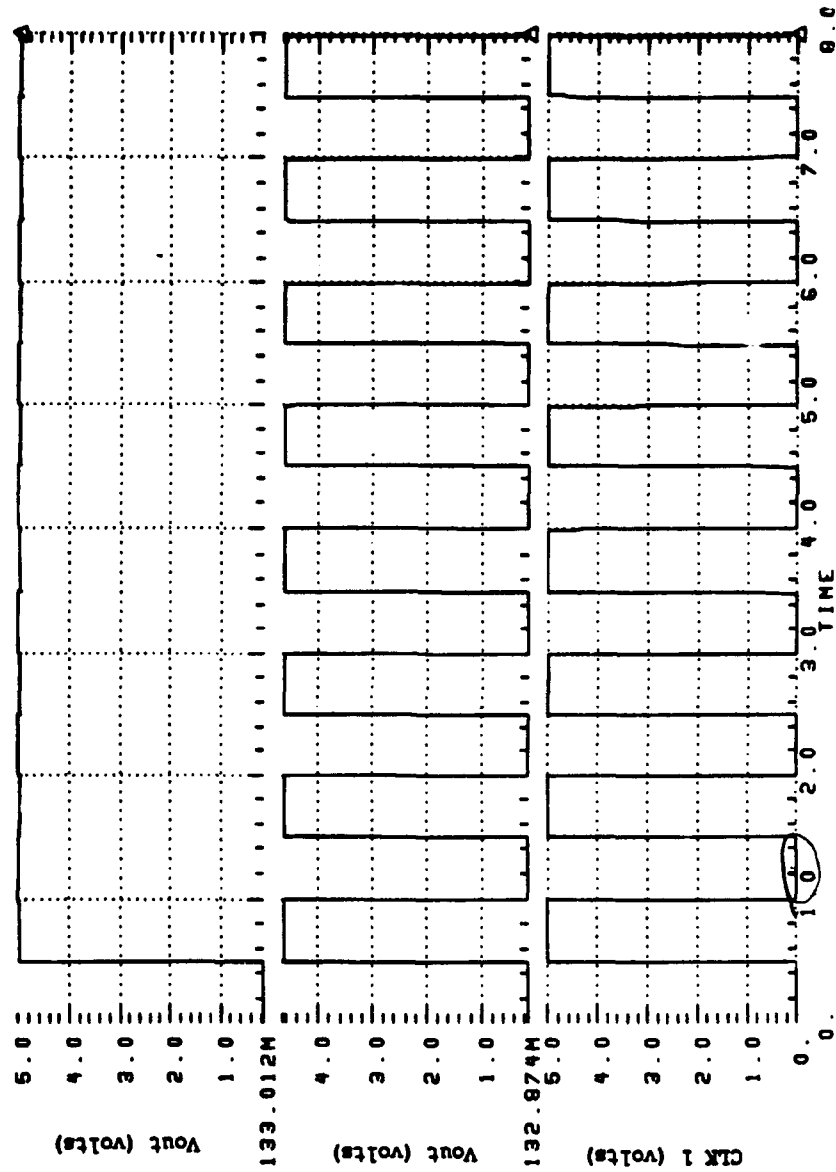


Figure 4.31: Clocked comparator's output when threshold voltage is greater than the input voltage.

voltage, then the comparator output will be low, or false as shown in Figure 4.32.

Once this comparator was simulated and found to operate correctly, the appropriate number of these comparators was connected to the modulus folding circuits and simulated. The initial spice simulation that was conducted was a transient analysis of a small segment of the dynamic range. The clocks were operating at a slow rate of 100Hz. As can be seen from the results in Figure 4.33, there were considerable oscillations, not only in the output of the comparator but also in the input which is the output of the modulus folding circuit. It appeared that the clock signals were being fed back into the modulus folding circuit. Even after some adjustments to the biasing of the comparator, the same phenomena continued.

Since this clocked comparator did not operate effectively, a new comparator was designed. Figure 4.34 is a schematic of this new design. Basically, this design consisted of a comparator, composed of a differential amplifier and an inverting output buffer, and a sample-and-hold circuit, comprised of a transmission gate and a capacitor. This comparator operated similarly to the clocked comparator. When the threshold voltage was larger than the input voltage, the output of the comparator would be high. However, when the threshold voltage was less than the voltage, the output of the comparator would be low. The

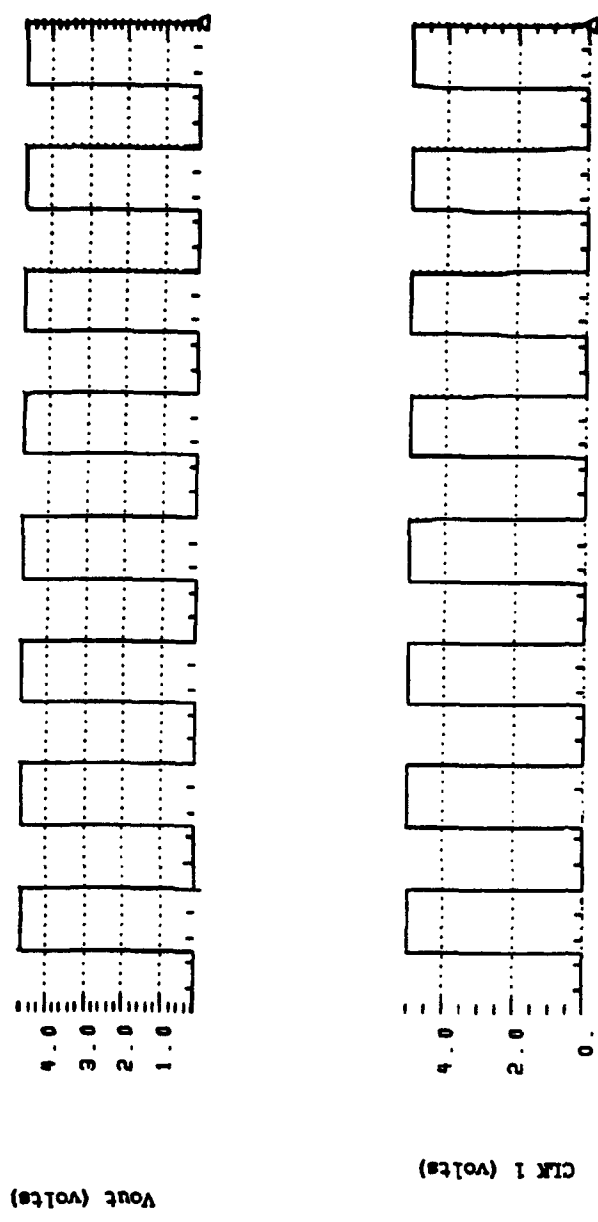
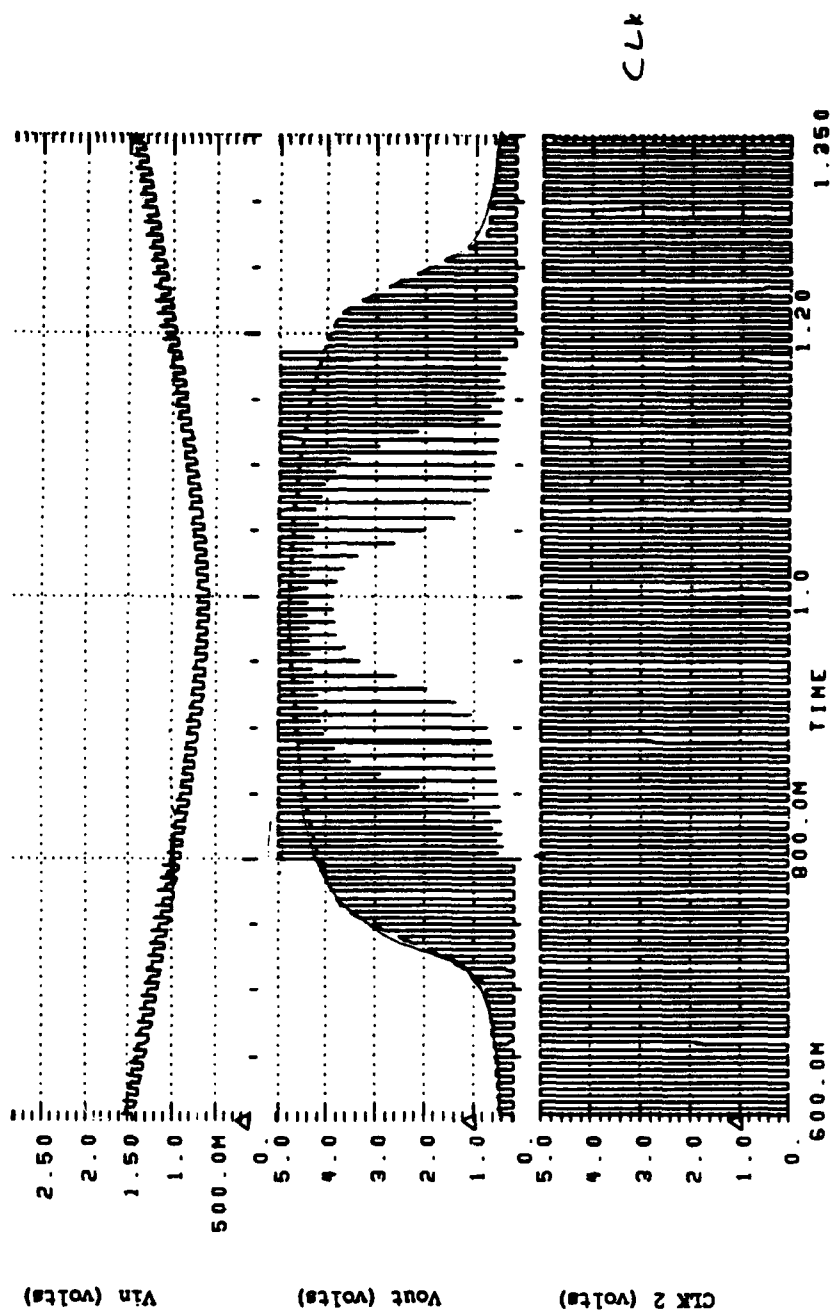


Figure 4.32: Clocked comparator's output when the threshold voltage is less than the input voltage.



CLK

Figure 4.33: Clocked comparator output with oscillations.

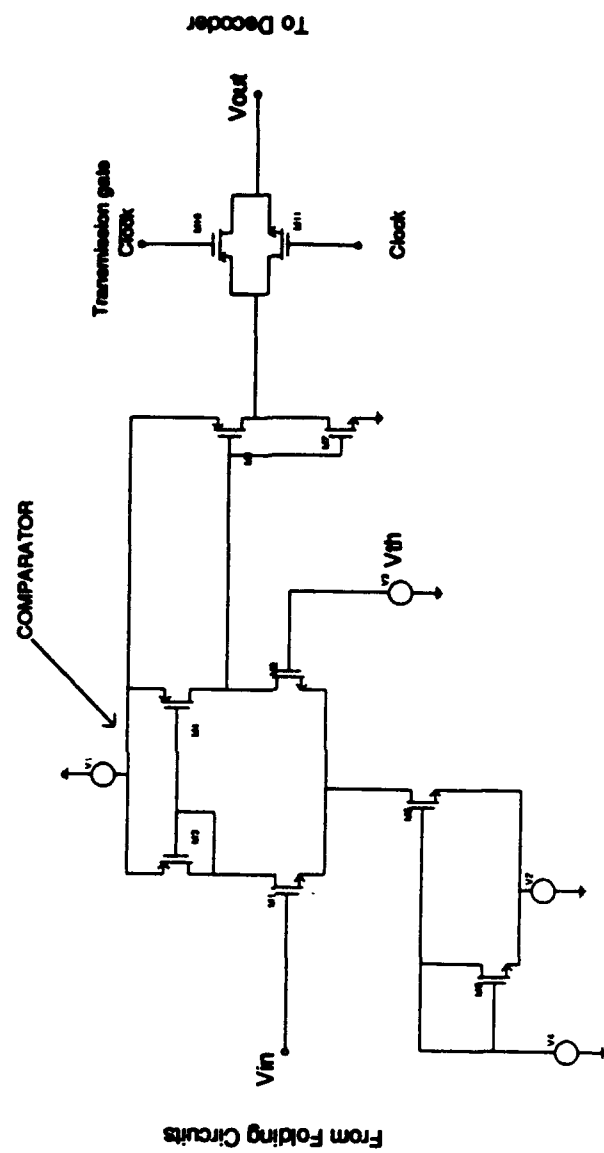


Figure 4.34: Static comparator with transmission gate.

sample-and-hold section of the comparator is controlled by two non-overlapping clocks, CLK1 and CLK2. When the CLK1 signal is high and the CLK2 signal is low, the transmission gate lets the signal pass through to charge the capacitor. When the CLK1 signal is low and the CLK2 signal is high, the transmission gate blocks the signal from passing through the gate, which allows the capacitor to retain its charge until CLK1 is high again (see Figure 4.35).

These comparators were connected to the modulus folding circuits, and spice transient analysis simulations were conducted. Figures 4.36-4.47 show the results of these simulations. It can be seen that the comparators were operating appropriately, going high when the output of the modulus folding circuit reaches the value of the predetermined threshold voltage. Appendix A is one of the spice input file which was used to conduct this simulation.

B. DIGITAL CIRCUIT DESIGN

After the modulus folding circuits and comparators were designed and simulated successfully, the design and simulation effort was turned to the digital subsystem. The digital subsystem consists of a decoder and a programmed logic array (PLA), as shown in Figure 1.8 and 3.3. The decoder is divided into two functions. The first of which is to transform the comparator output values of each modulus into a binary thermometer format. Table 4.7 depicts the truth table, which

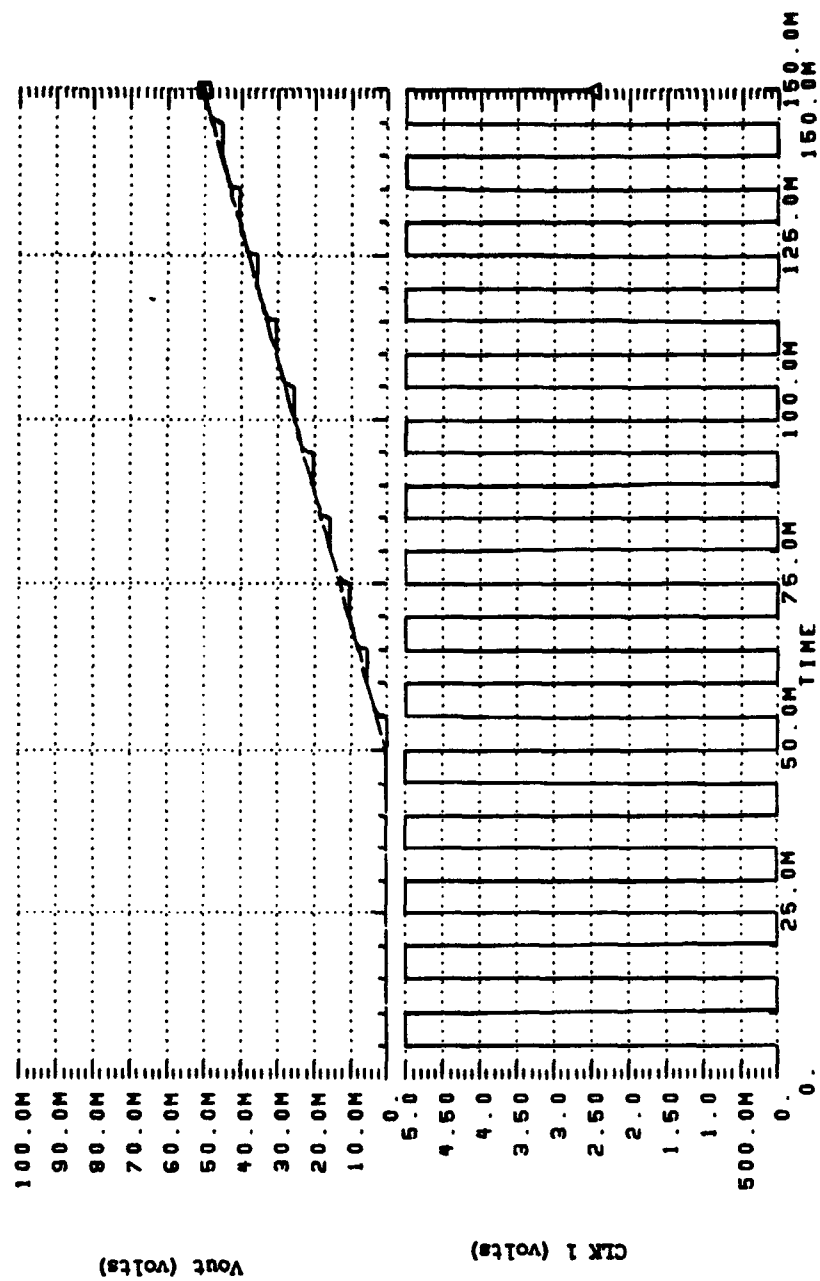


Figure 4.35: Operation of a transmission gate.

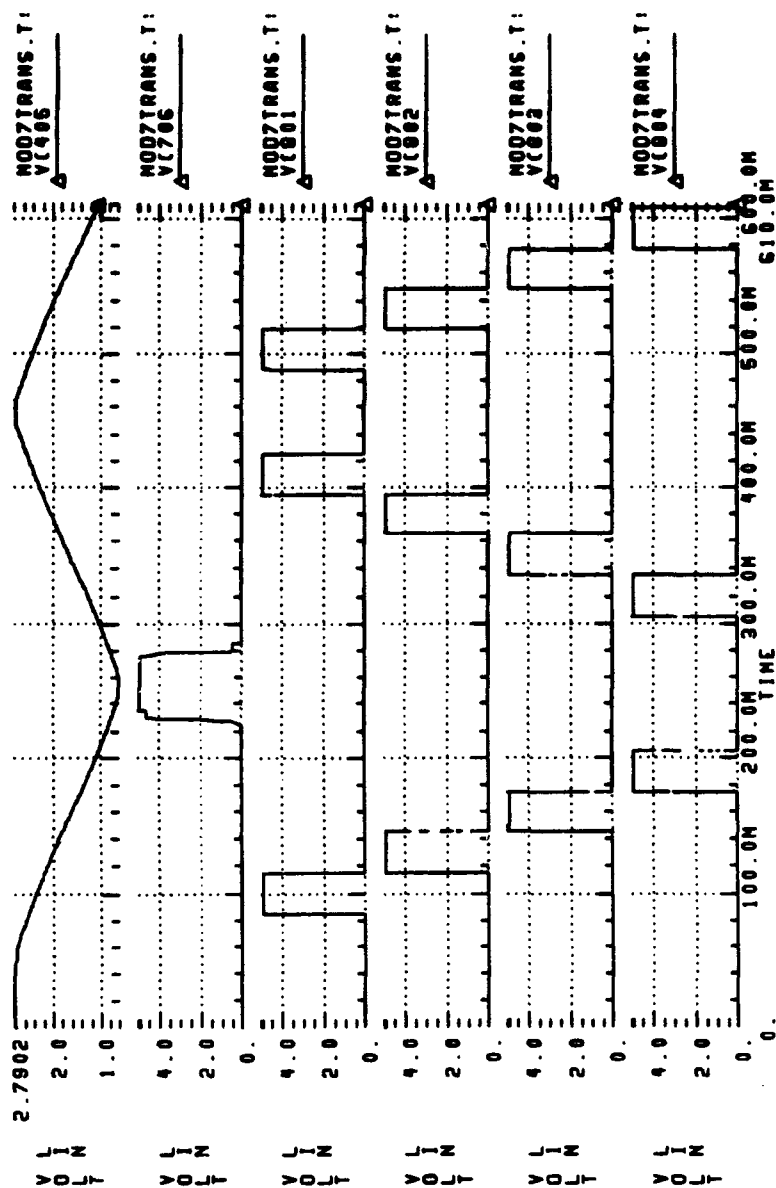


Figure 4.37: Mod 7 comparator and 1st decoder outputs.

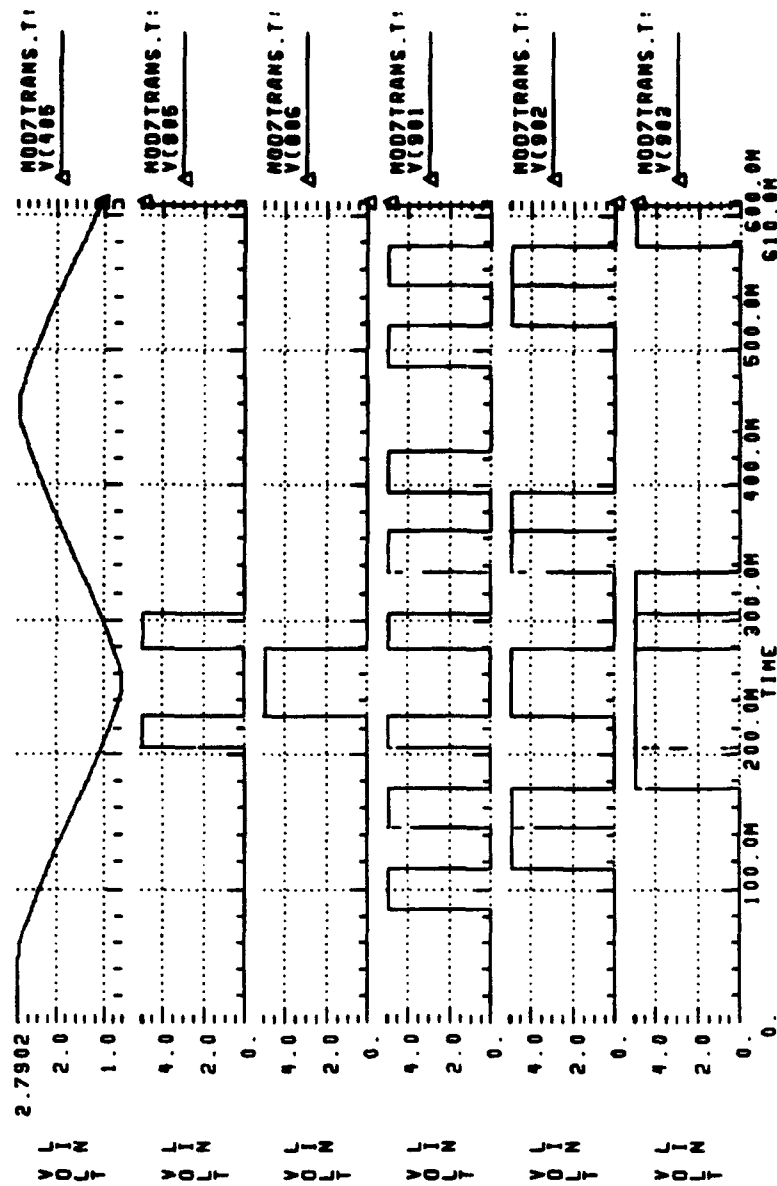


Figure 4.38: Mod 7 1st and 2nd decoder outputs.

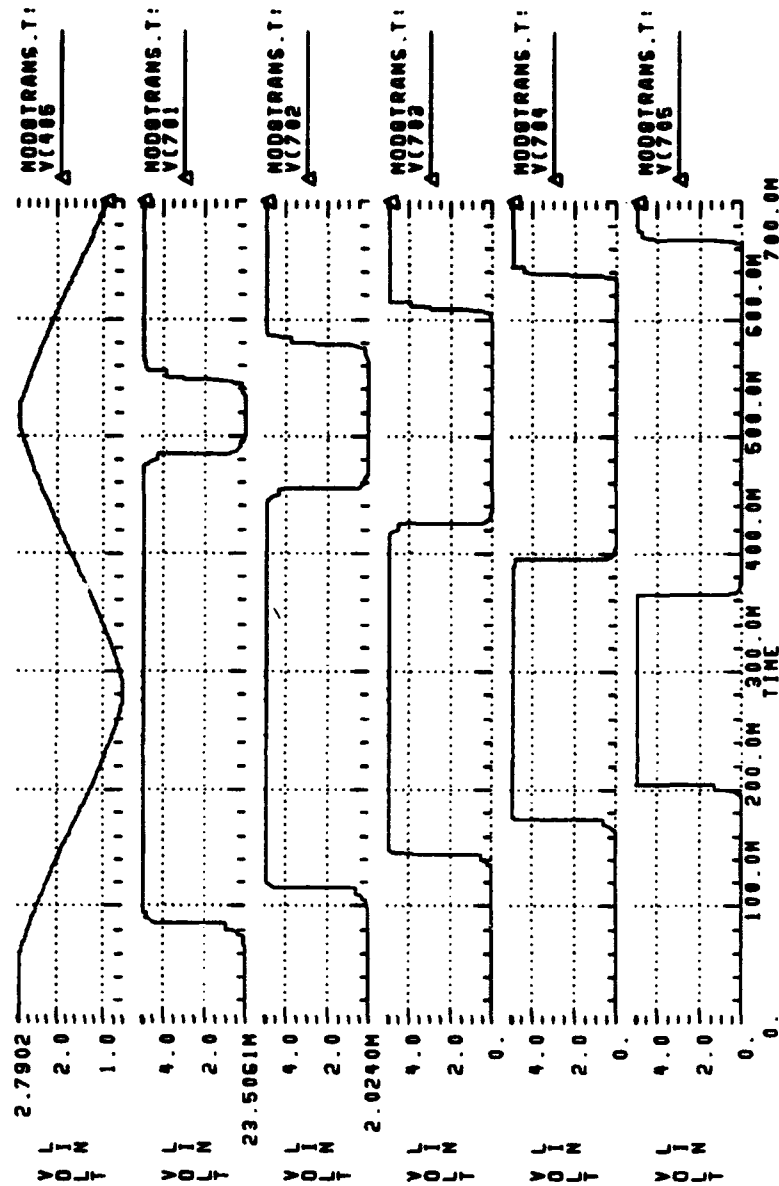


Figure 4.39: Mod 8 comparator outputs.

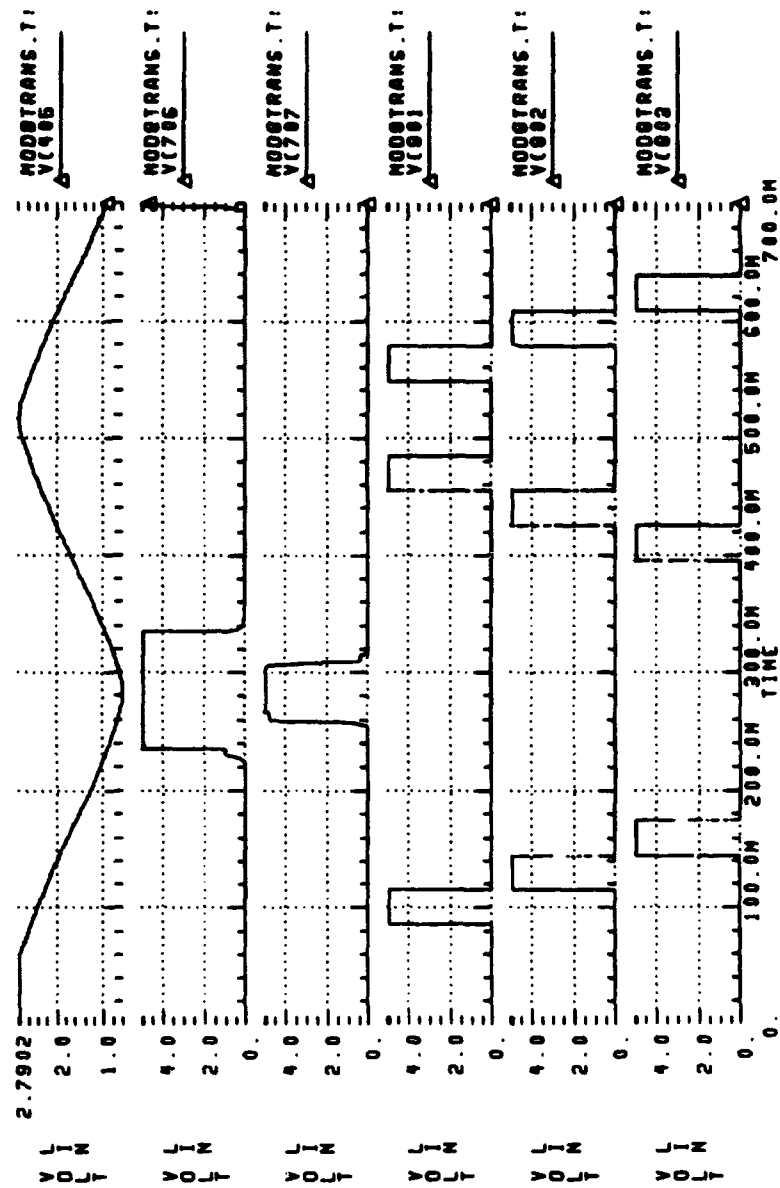


Figure 4.40: Mod 8 comparator and 1st decoder outputs.

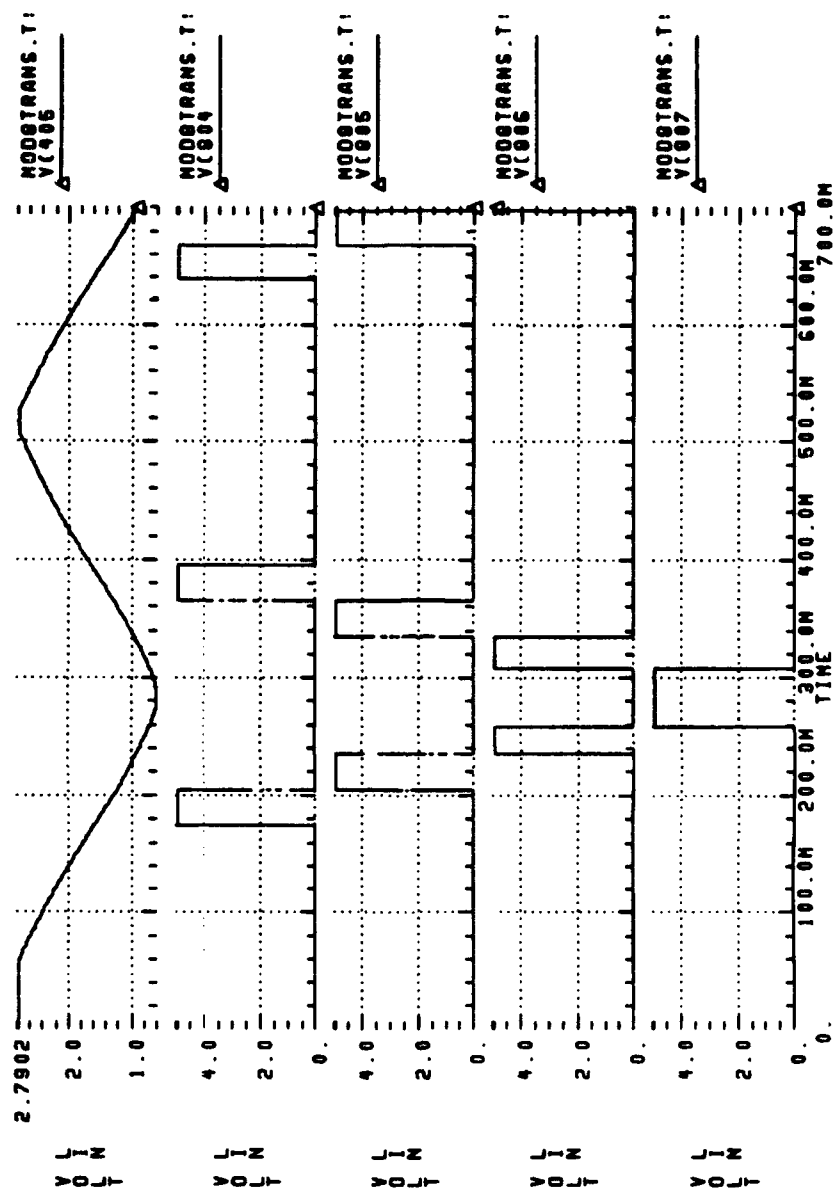


Figure 4.41: Mod 8 1st decoder outputs.

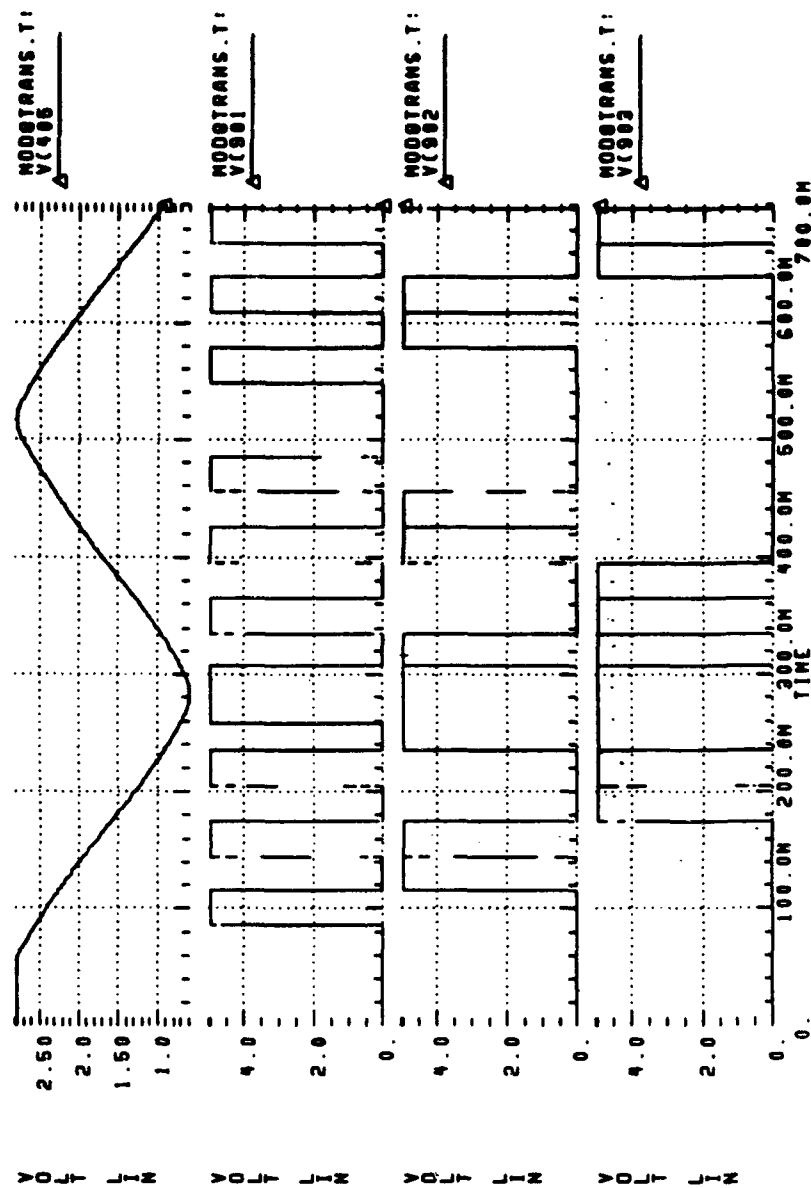


Figure 4.42: Mod 8 2nd decoder output.

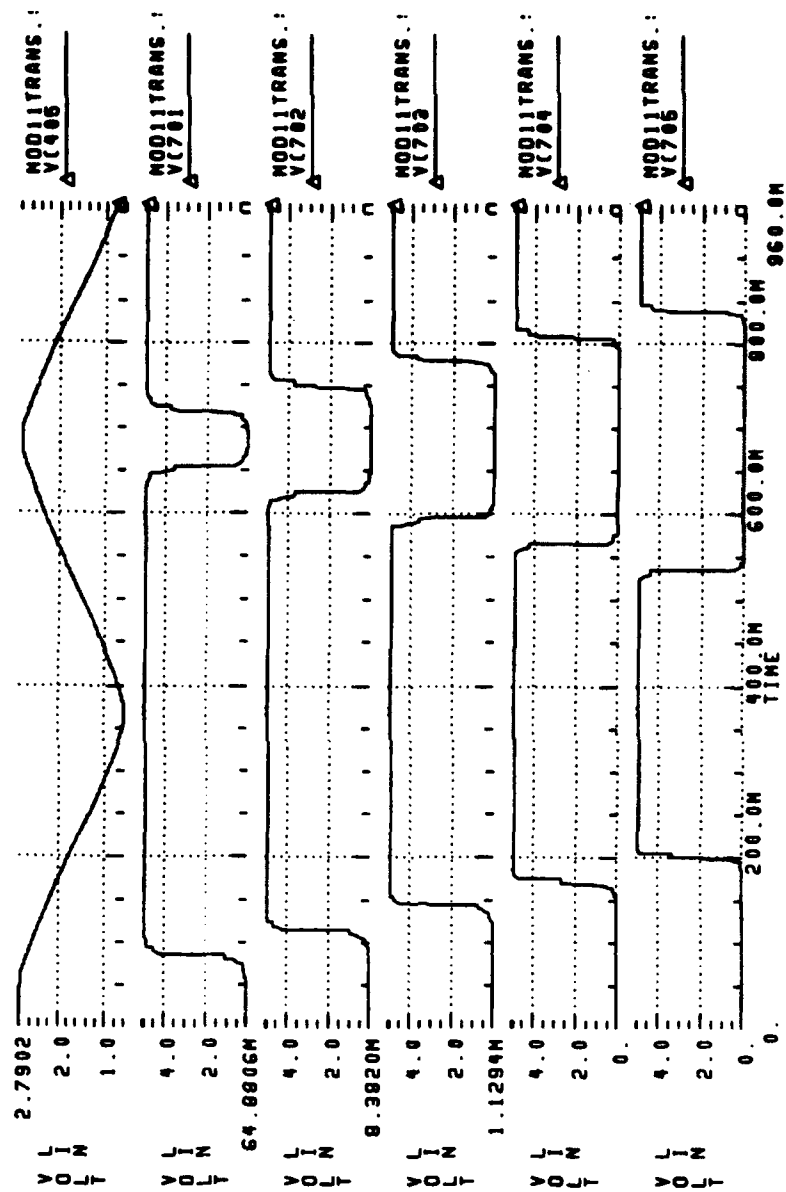


Figure 4.43: Mod 11 comparator outputs.

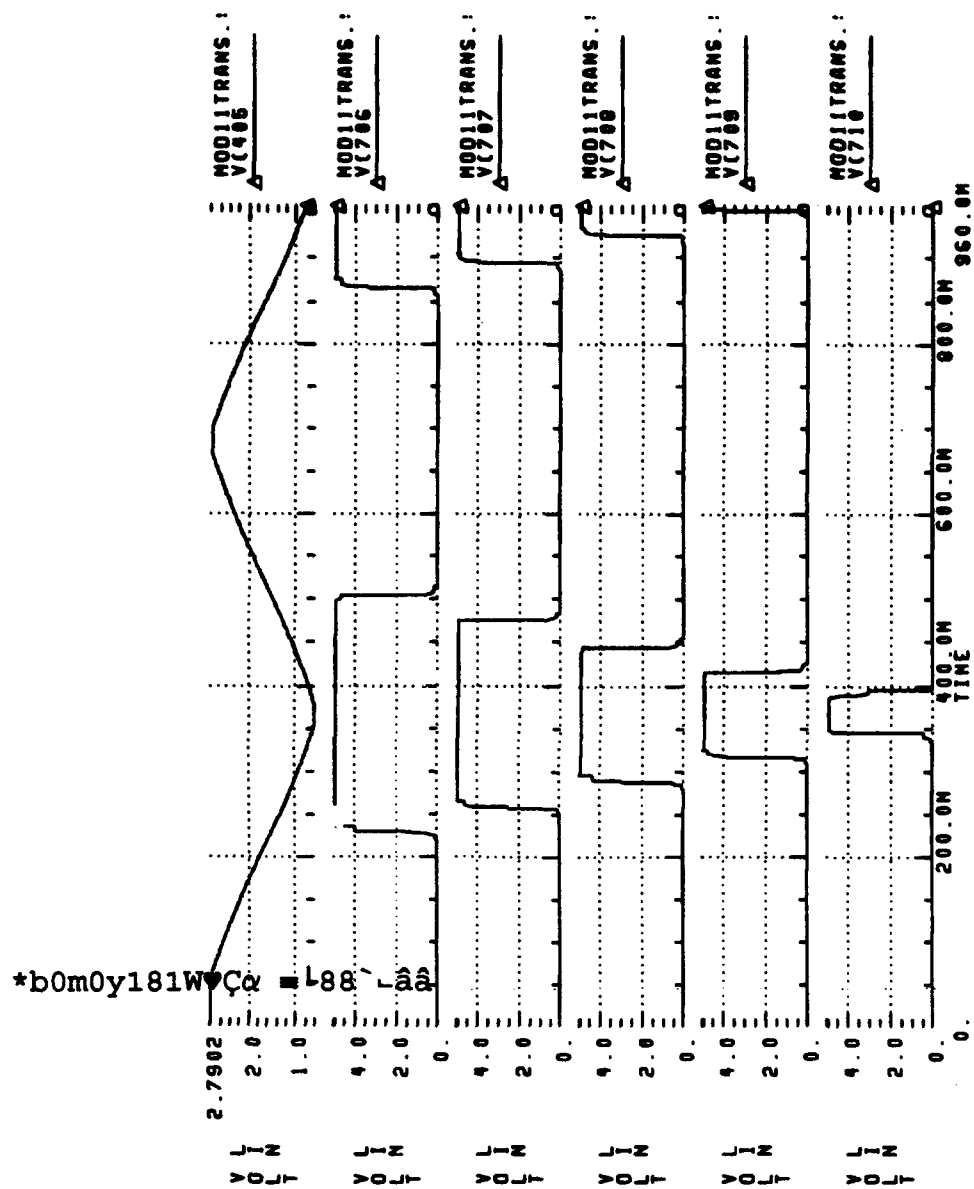


Figure 4.44: Mod 11 comparator outputs continued.

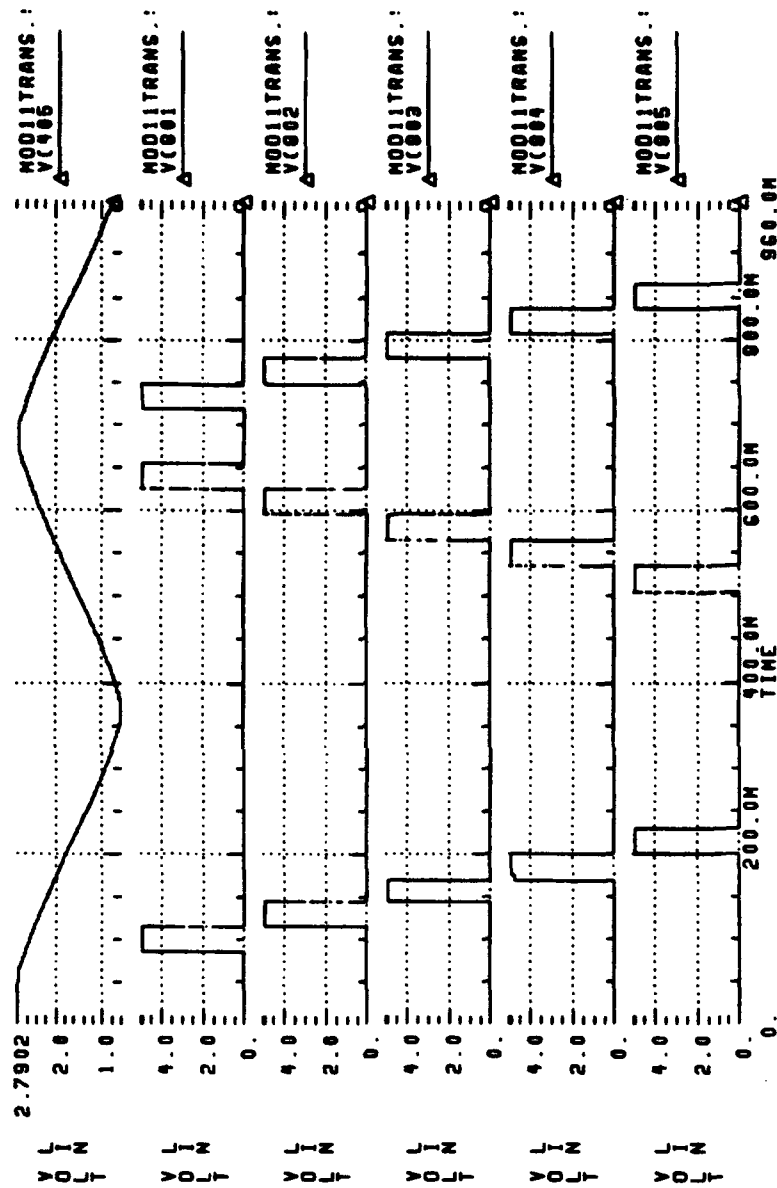


Figure 4.45: Mod 11 1st decoder output.

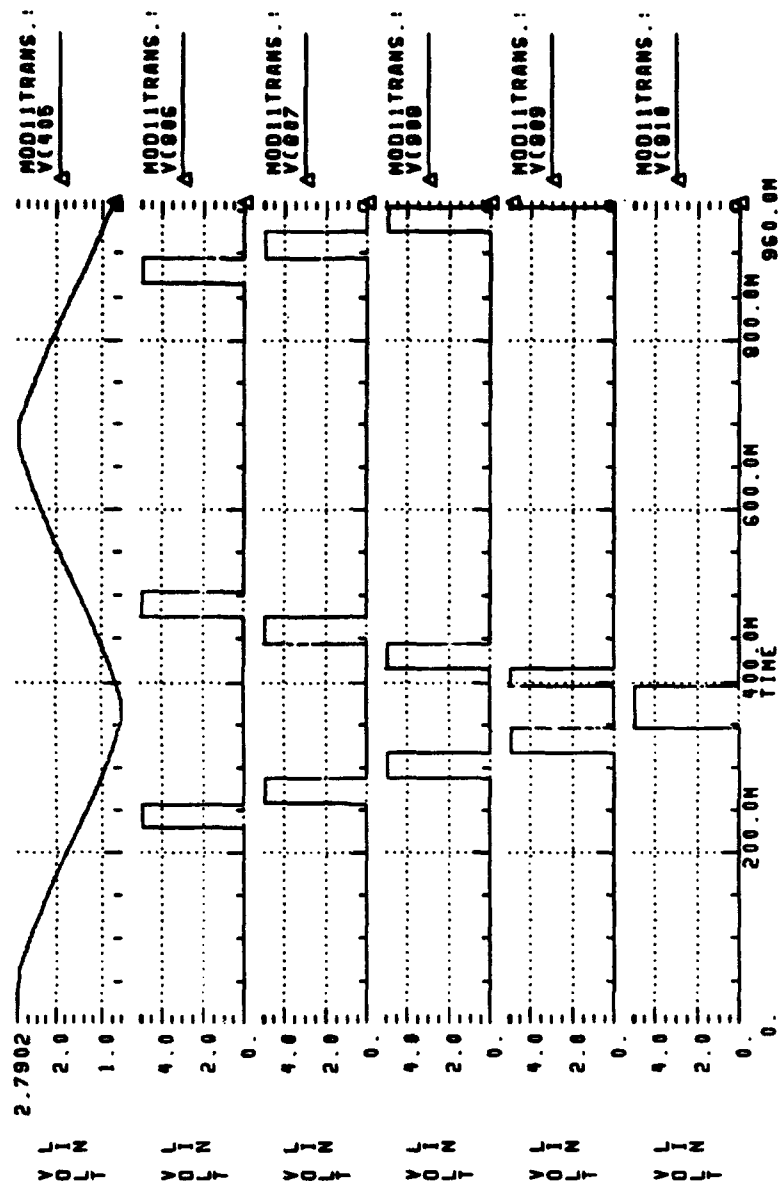


Figure 4.46: Mod 11 1st decoder output continued.

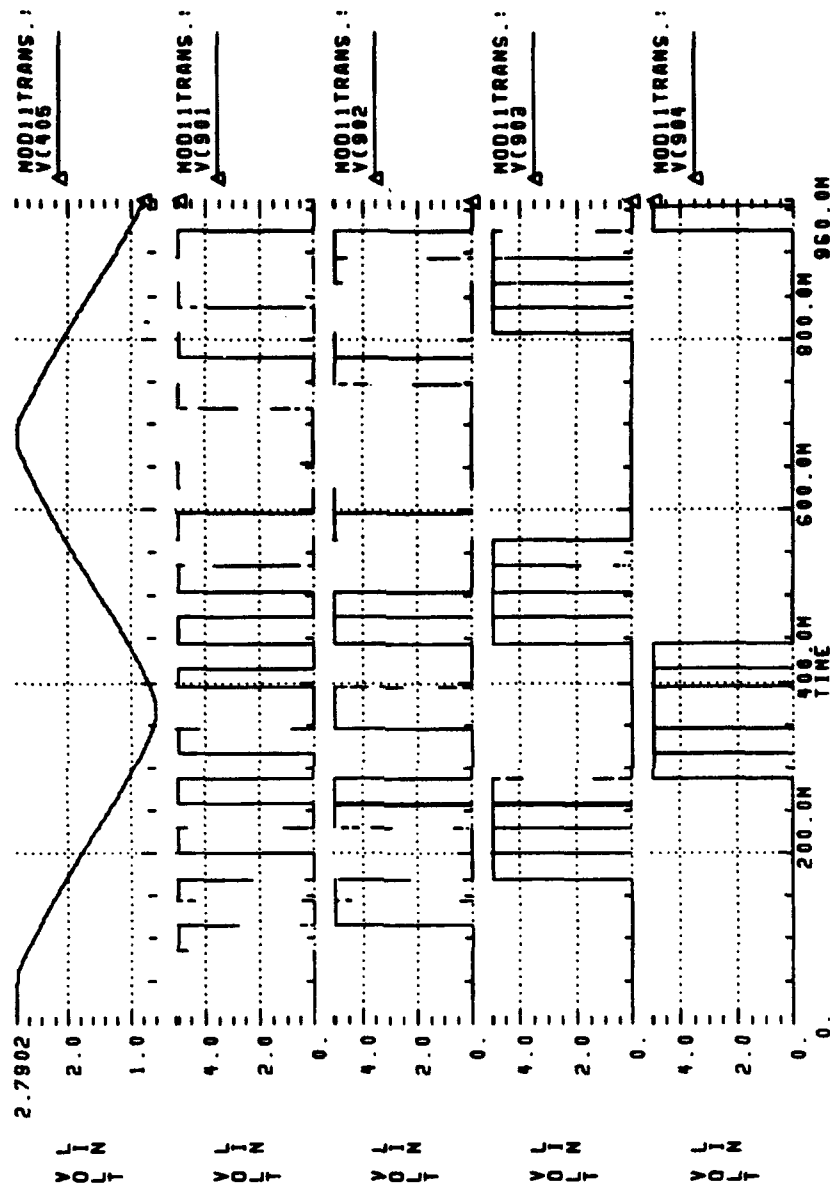


Figure 4.47: Mod 11 2nd decoder output.

illustrates this function for a modulus 7 folding circuit comparator output. For an input from the comparators of 111111, the decoder would translate it as an output of 100000. This functionality would be applied to the other moduli, specifically modulus 8 and modulus 11. Figure 4.48 is the schematics of the digital circuit which implement the logic presented in Table 4.7. The logic gates are implemented using CMOS technology and are shown in Figure 4.49. This part of the decoder was simulated and the results are shown in Figures 4.50 and 4.51, validating this circuit design.

The second function of the decoder, as depicted in Figure 4.52 for mod 7, is to take the thermometer coded output of the first part of the decoder and translate it to a straight binary number. This number is the binary value of the number of comparators that are on, or high at a given moment. As an example, Table 4.8 illustrates the functionality of this part of the decoder in a truth table format for the modulus 7 subcircuit. If the input is 100000, then the output should be 110. Figure 4.53 shows the simulation of the circuit of Figure 4.52. The digital circuits for the other moduli can be readily extracted from the mod 7 design. When the output of each of the modulus decoders are integrated, this represents the analog input voltage of the Analog-to-Digital Converter in the SNS format. As can be seen in the simulations in Figures 4.36-4.47, this decoder worked very well for each of the modulus subcircuits. As the input voltage is swept through

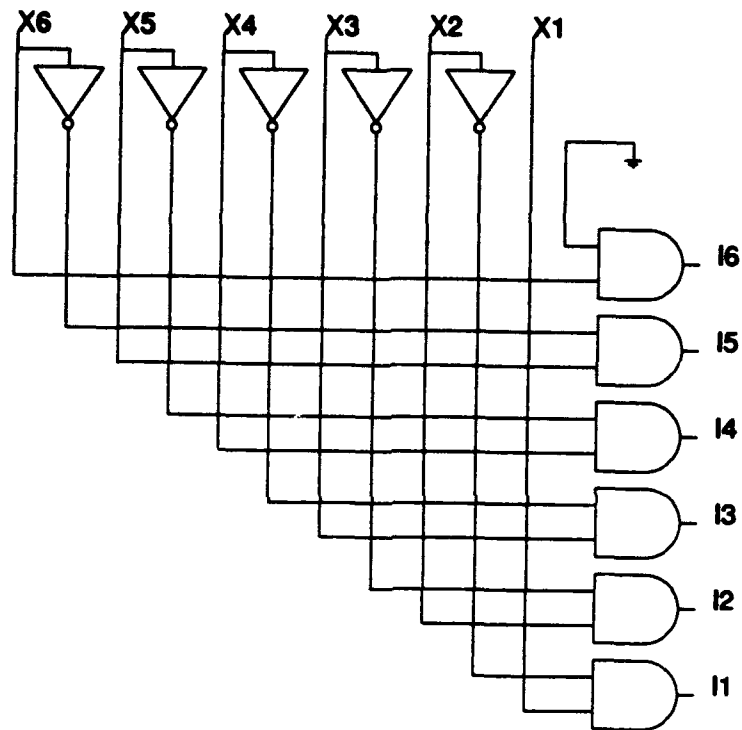
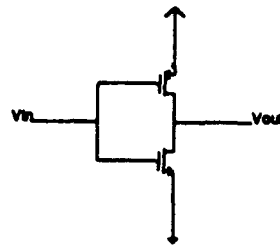
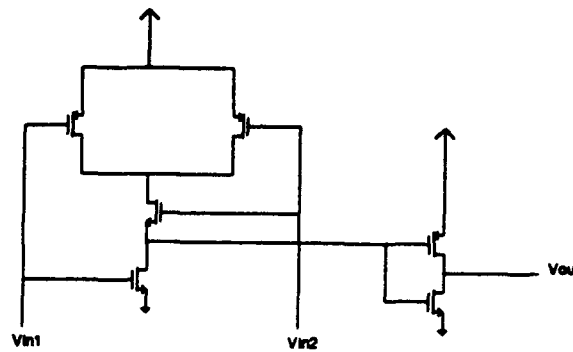


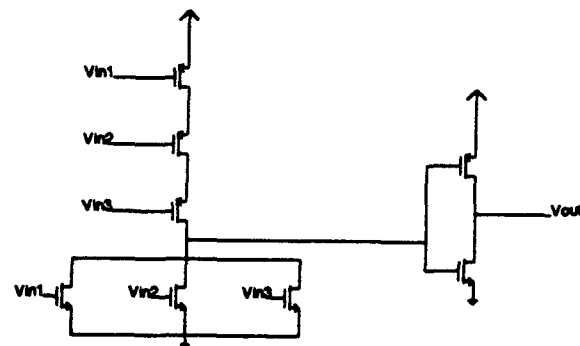
Figure 4.48: First half of the mod 7 decoder.



(a) Inverter



(b) 2-input AND gate



(c) 3-input OR gate

Figure 4.49: CMOS gates used in decoder design.

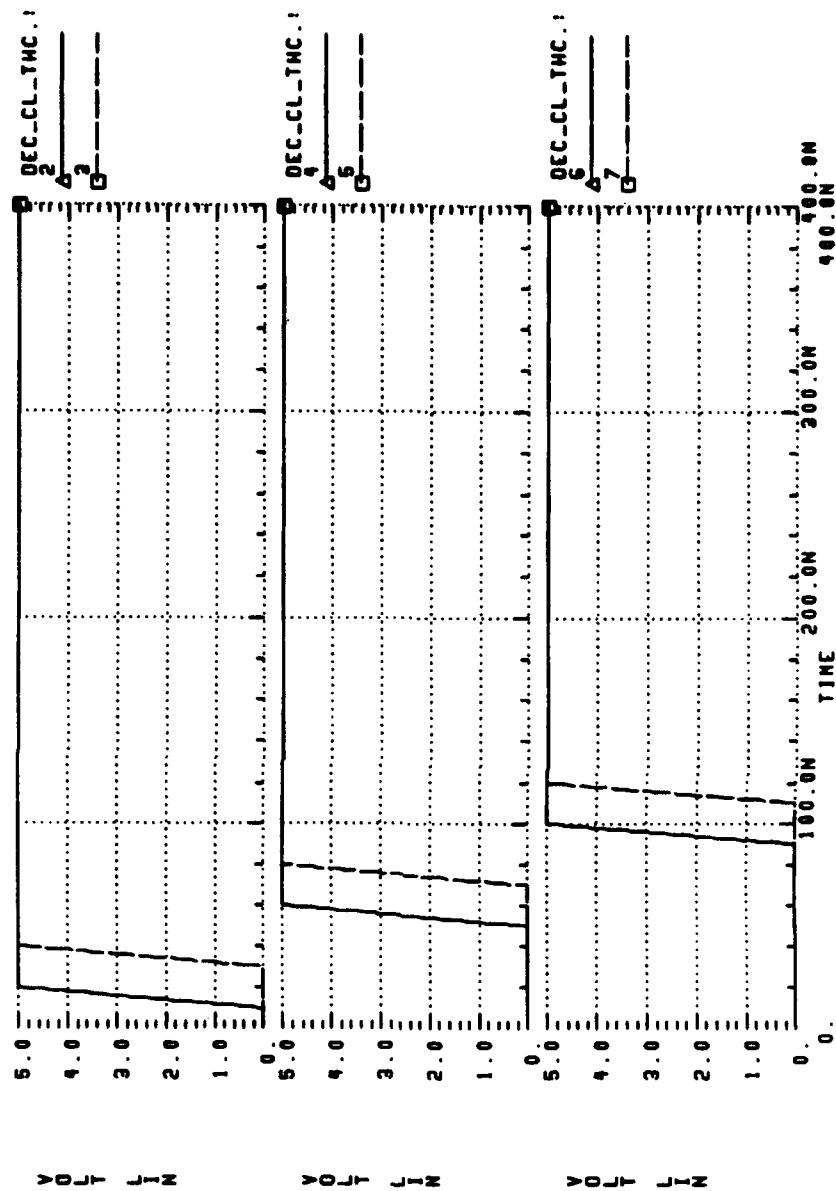


Figure 4.50: 1st decoder simulation showing the input comparator levels.

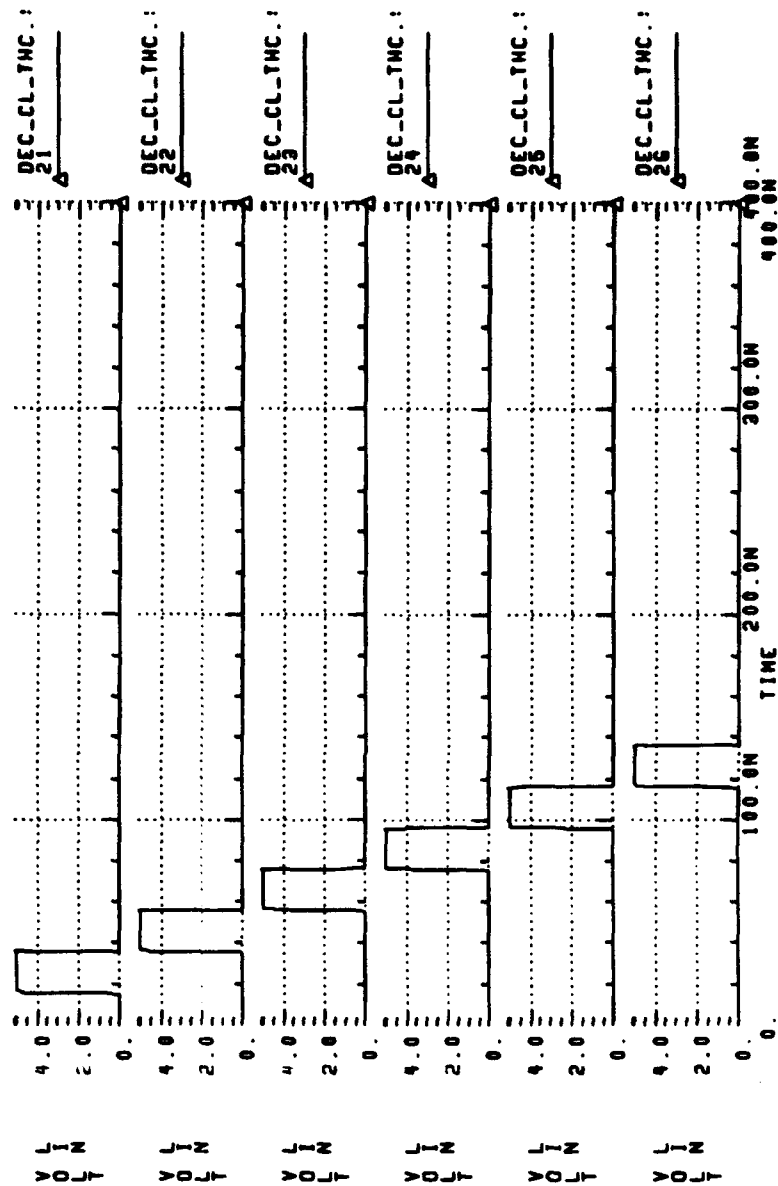


Figure 4.51: 1st decoder simulation showing decoder output.

TABLE 4.7 1ST DECODER TRUTH TABLE

I ₆	I ₅	I ₄	I ₃	I ₂	I ₁		Y ₆	Y ₅	Y ₄	Y ₃	Y ₂	Y ₁
0	0	0	0	0	0		0	0	0	0	0	0
0	0	0	0	0	1		0	0	0	0	0	1
0	0	0	0	1	1		0	0	0	0	1	0
0	0	0	1	1	1		0	0	0	1	0	0
0	0	1	1	1	1		0	0	1	0	0	0
0	1	1	1	1	1		0	1	0	0	0	0
1	1	1	1	1	1		1	0	0	0	0	0

TABLE 4.8 2ND DECODER TRUTH TABLE

I ₆	I ₅	I ₄	I ₃	I ₂	I ₁		Y ₃	Y ₂	Y ₁
0	0	0	0	0	0		0	0	0
0	0	0	0	0	1		0	0	1
0	0	0	0	1	0		0	1	0
0	0	0	1	0	0		0	1	1
0	0	1	0	0	0		1	1	0
0	1	0	0	0	0		1	1	1
1	0	0	0	0	0		1	1	0

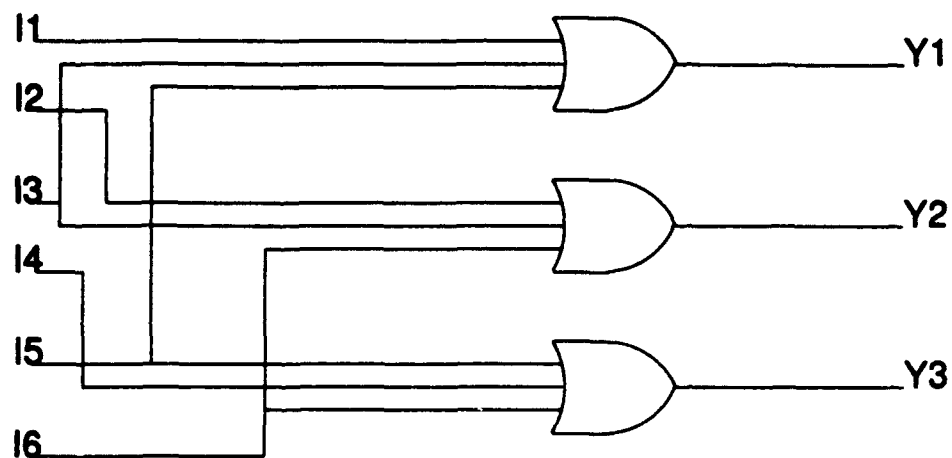


Figure 4.52: Second half of the mod 7 decoder.

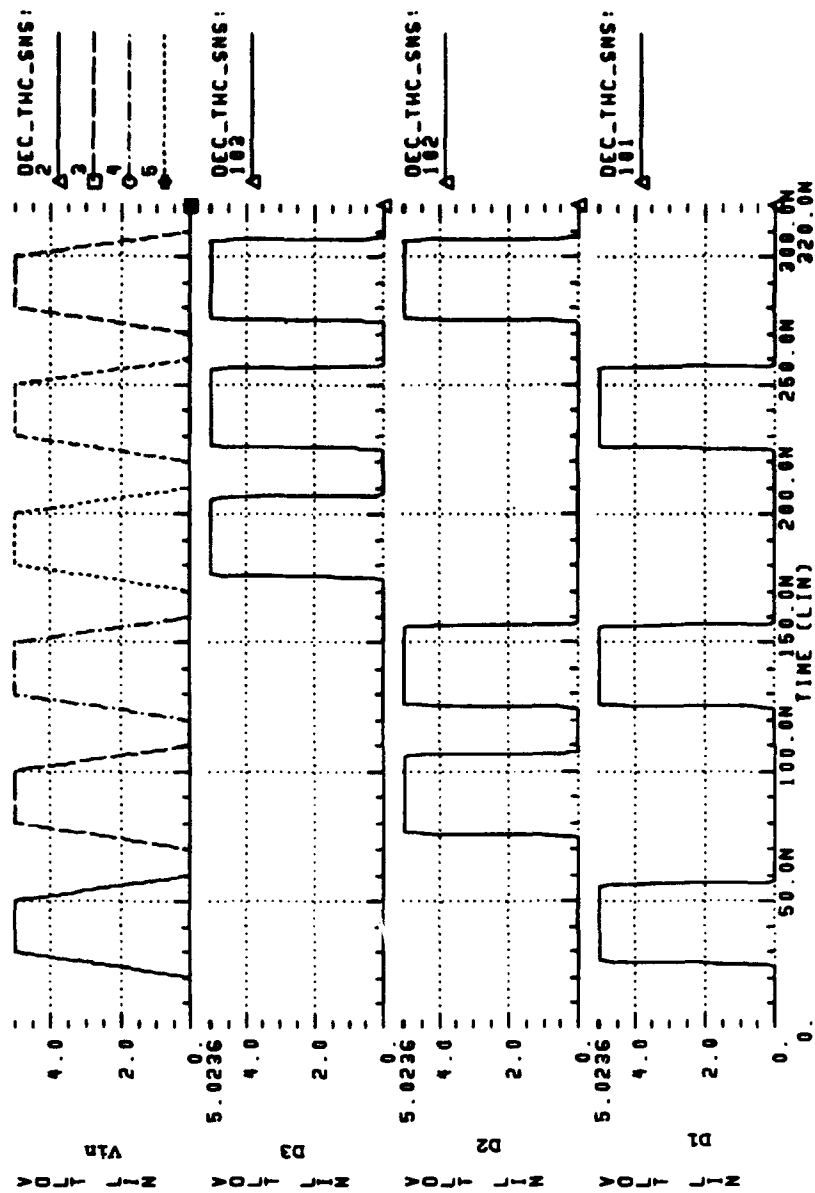


Figure 4.53: 2nd decoder simulation showing the decoder output.

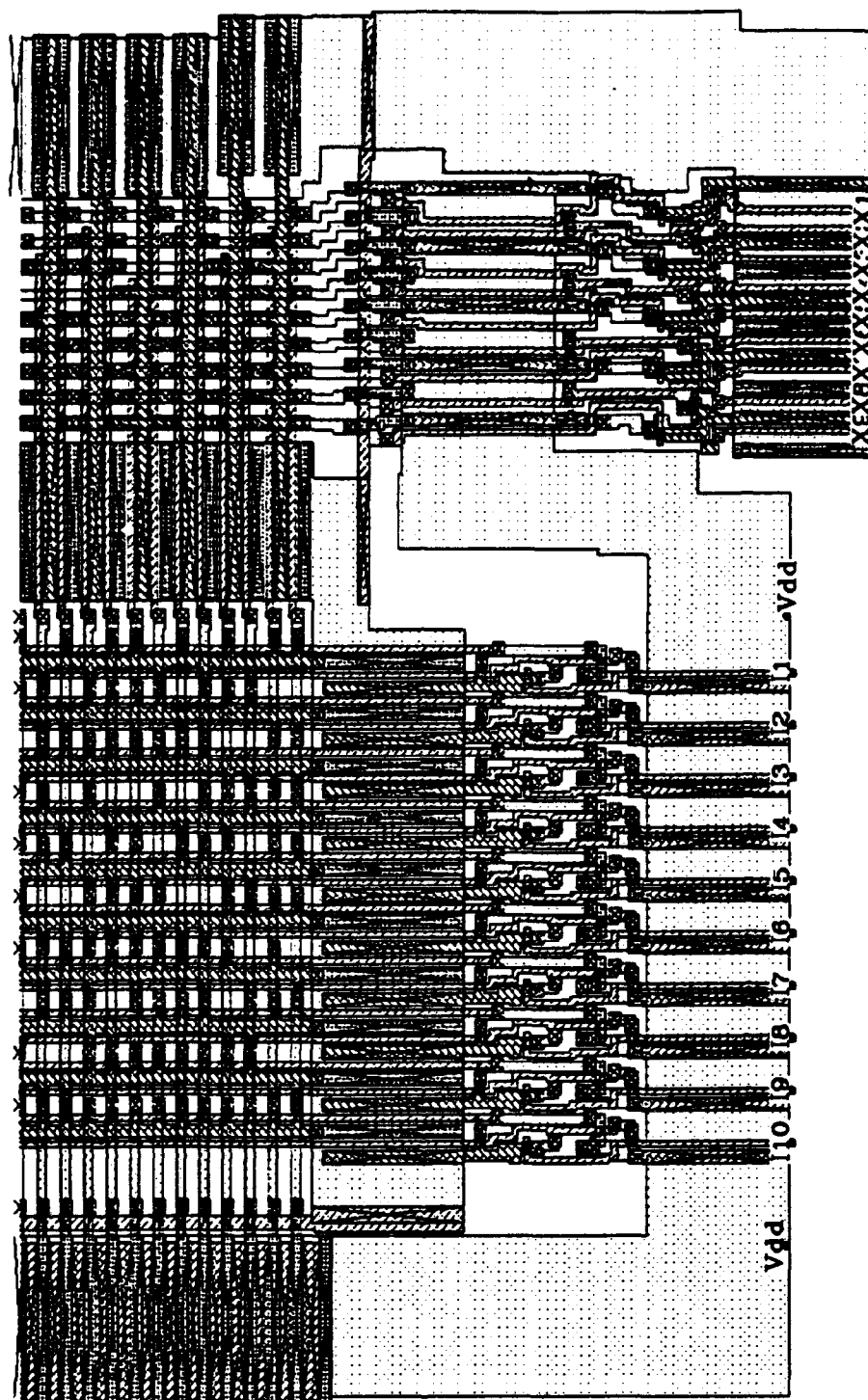


Figure 4.54: Partial layout of a PLA.

its dynamic range, each individual modulus subcircuit can be seen stepping through its modulus sequence. For instance, in the modulus 7 subcircuit, the output can be seen to go through the sequence: 000, 001, 010, 011, 100, 101, 110.

The output of the decoder represents the SNS value of the input signal applied to the overall analog-to-digital converter. This output is then fed to a PLA which converts it to a standard binary format. The number of inputs into the PLA would be ten, and for a 9-bit analog-to-digital converter the number of outputs for the PLA would be 9. The design of the PLA is virtually an automatic process. Using U.C. Berkeley CAD tools, all that is needed is to specify the PLA inputs and outputs in a file and then run a PLA generation program to produce a CMOS layout partially shown in Figure 4.54. The input file can be found in Appendix B, but the actual PLA truth table was generated by a MATLAB program which is presented in Appendix C. Information from the PLA layout was extracted to perform digital simulation to investigate the validity of the design (see Appendix D). Once this validation is complete, the PLA would then be connected to the rest of the analog-to-digital converter so that a spice simulation can be conducted of the entire analog-to-digital converter.

V. STRENGTHS, LIMITATIONS, AND FUTURE DEVELOPMENTS

The greatest contribution or strength of the 9-bit analog-to-digital design presented in this thesis, is the reduction of the number of comparators. In a standard flash ADC, the number of comparators required would be $2^9-1=511$. In the design presented in this thesis, the number of comparators that were used is 23. This is a dramatic reduction which translates to a decrease in power consumption and to smaller chip size. The reduction in the number of comparators should theoretically improve the speed of the ADC. However, it was found that the design could only achieve a maximum frequency in the range of 100kHz instead of the expected Mhz range which have been achieved in other research. This thesis presented data which indicated that increases in the bias current of the folding circuits did improve the frequency response. However, the drawback was the widening of the width of the folded output of the modulus folding circuits. This resulted in destroying the functionality and periodicity of the folded output to implement the preprocessing architecture.

Another area of limitation is the resolution. This thesis initially envisioned a 10-bit ADC but had to be reduced to a 9-bit resolution because of the gain limitation of the CMOS process that was used. In order to achieve a 10-bit resolution, the width of the MOS transistors were

impractically large to achieve the proper width in the folded output of the folding circuits. It is believed that a new process with favorable gain characteristics could provide a better frequency response and enable higher resolution.

Future research endeavors should use a more suitable CMOS process and the results of this thesis to produce a CMOS layout and a physical application specific integrated chip (ASIC). Perhaps, when these steps are achieved a GaAs ADC can be built utilizing the information presented in this thesis to achieve even better performance.

APPENDIX A

SPICE INPUT FILE

CMOS MOD 7 FOLDING CIRCUIT FOR 9-BIT ADC

* SPICE3C PMOS and NMOS model level 2 corner parameters for
* the VLSI Technology 2.0U CMOS N-well process. 10/23/92
* SPICE3C PMOS and NMOS model level 2 nominal parameters,
* derived from corner parameters above.

.MODEL npf PMOS (LEVEL=2 VTO=-0.75 TOX=400E-10 NSUB=6.0E+15
+ XJ=0.05U LD=0.20U U0=255 UCRIT=0.86E5 UEXP=0.29
+VMAX=3.0E4 + NEFF=2.65 DELTA=1.0 RSH=101 CGSO=1.9E-10
+CGDO=1.9E-10
+ CJ=250U CJSW=350P MJ=0.535 MJSW=0.34 PB=0.8)

.MODEL nnf NMOS (LEVEL=2 VTO=+0.775 TOX=400E-10 NSUB=8.0E+15
+ XJ=0.15U LD=0.20U U0=650 UCRIT=0.62E5 UEXP=0.125
+VMAX=5.1E4 + NEFF=4.0 DELTA=1.4 RSH=36 CGSO=1.95E-10
+CGDO=1.95E-10
+ CJ=195U CJSW=500P MJ=0.76 MJSW=0.30 PB=0.8)

*Sources

VDD 1 0 18V
VSS 2 0 -18V
*Is 4 0 .5mA
Vop+ 7 0 15V
vop- 8 0 -15V
VC+ 500 0 5.0V
VC- 501 0 -5.0V

*Input Resistors for Op Amp

R1 211 301 1K
R2 212 301 1K
R3 213 301 1K
R4 214 301 1k
R5 215 301 1K
R6 216 301 1K
R7 217 301 1K
R8 218 301 1K
R9 219 301 1K
R10 220 301 1K
R11 221 301 1K
R12 222 301 1K
R13 223 301 1K

```

R14  224 301 1k
R15  225 301 1K
R16  226 301 1K
R17  227 301 1K
R18  228 301 1K
R19  229 301 1K
R20  230 301 1K
R21  231 301 1K
R22  232 301 1K
R23  233 301 1K
R24  234 301 1k
R25  235 301 1K
R26  236 301 1K
R27  237 301 1K
R28  238 301 1K
R29  239 301 1K
R30  240 301 1K
R31  241 301 1K
R32  242 301 1K
R33  243 301 1K
R34  244 301 1k
R35  245 301 1K
R36  246 301 1K
R37  247 301 1K
R38  248 301 1K

```

*Feedback Resistor for the Op Amp

```
RF  306 301 20K
```

*Signals

```
*Vin  3 0
```

```
*Vin  3 0 PWL(0,0.0 10us,4.0)
```

```
*Vin  3 0 PWL(0,0.0 10000ms,4.0)
```

```
*Vin  3 0 PWL(0,0 4ns,4)
```

```
Vin  3 0 PWL 0 0.0V, 40u 15.1V td=32.67n
```

*Clocks for Comparators

```
*VCLK2 502 0 pulse( 0v 5v .99s .01s .01s .49s 1s)
```

```
*VCLK2 502 0 pulse( 0v 5v .099s .001s .001s .049s .1s)
```

```
*VCLK2 502 0 pulse( 0v 5v .0099s .0001s .0001s .0049s
```

```
.01s) *VCLK2 502 0 pulse( 0v 5v .00495s .00005s .00005s
```

```
.00245s .005s) VCLK2 502 0 pulse( 0v 5v 0.0s .0001s .0001s
```

```
.0049s .01s) *VCLK1 503 0 pulse(0v 5V .49s .01s .01s .49s
```

```
1s)
```

```
*VCLK1 503 0 pulse(0v 5V .049s .001s .001s .049s .1s)
```

```
*VCLK1 503 0 pulse(0v 5V .0049s .0001s .0001s .0049s
```

```
.01s) *VCLK1 503 0 pulse(0v 5V .00245s .00005s .00005s
```

```
.00245s .005s) VCLK1 503 0 pulse(0v 5V .005s .0001s .0001s
```

```
.0049s .01s)
```

*Reference Voltages

V11	11	0	0.0V
V12	12	0	.41V
V13	13	0	.82V
V14	14	0	1.23V
v15	15	0	1.64V
V16	16	0	2.05V
v17	17	0	2.46V
V18	18	0	2.87V
V19	19	0	3.28V
V20	20	0	3.69V
V21	21	0	4.10V
V22	22	0	4.51V
V23	23	0	4.92V
V24	24	0	5.33V
v25	25	0	5.74V
V26	26	0	6.15V
v27	27	0	6.56V
V28	28	0	6.97V
V29	29	0	7.38V
V30	30	0	7.79V
V31	31	0	8.20V
V32	32	0	8.61V
V33	33	0	9.02V
V34	34	0	9.43V
v35	35	0	9.84V
V36	36	0	10.25V
V37	37	0	10.66V
V38	38	0	11.07V
V39	39	0	11.48V
V40	40	0	11.89V
V41	41	0	12.30V
V42	42	0	12.71V
V43	43	0	13.12V
V44	44	0	13.53V
V45	45	0	13.94V
V46	46	0	14.35V
V47	47	0	14.76V
V48	48	0	15.17V

***Comparator Reference Voltages**

V601	601	0	2.58V
v602	602	0	2.27V
V603	603	0	1.92V
V604	604	0	1.54V
V605	605	0	1.16V
V606	606	0	0.858V

***Folding Ckts external loads**

*M5	5	5	1	1	npf	L=2u	W=250u
*M6	6	6	1	1	npf	L=2u	W=250u

***Folding Circuit instances**

X1	1	2	3	111	1	1	11	fold1
X2	1	2	3	112	1	1	12	fold2
X3	1	2	3	113	1	1	13	fold3
X4	1	2	3	114	1	1	14	fold4
X5	1	2	3	115	1	1	15	fold5
X6	1	2	3	116	1	1	16	fold6
X7	1	2	3	117	1	1	17	fold7
X8	1	2	3	118	1	1	18	fold8
X9	1	2	3	119	1	1	19	fold9
x10	1	2	3	120	1	1	20	fold10
X11	1	2	3	121	1	1	21	fold11
X12	1	2	3	122	1	1	22	fold12
X13	1	2	3	123	1	1	23	fold13
X14	1	2	3	124	1	1	24	fold14
X15	1	2	3	125	1	1	25	fold15
X16	1	2	3	126	1	1	26	fold16
X17	1	2	3	127	1	1	27	fold17
X18	1	2	3	128	1	1	28	fold18
X19	1	2	3	129	1	1	29	fold19
x20	1	2	3	130	1	1	30	fold20
X21	1	2	3	131	1	1	31	fold21
X22	1	2	3	132	1	1	32	fold22
X23	1	2	3	133	1	1	33	fold23
X24	1	2	3	134	1	1	34	fold24
X25	1	2	3	135	1	1	35	fold25
X26	1	2	3	136	1	1	36	fold26
X27	1	2	3	137	1	1	37	fold27
X28	1	2	3	138	1	1	38	fold28
X29	1	2	3	139	1	1	39	fold29
x30	1	2	3	140	1	1	40	fold30
X31	1	2	3	141	1	1	41	fold31
X32	1	2	3	142	1	1	42	fold32
X33	1	2	3	143	1	1	43	fold33
X34	1	2	3	144	1	1	44	fold34
X35	1	2	3	145	1	1	45	fold35
X36	1	2	3	146	1	1	46	fold36
X37	1	2	3	147	1	1	47	fold37
X38	1	2	3	148	1	1	48	fold38

***Emitter follower Circuit instances**

X101	1	2	111	211	emit
X102	1	2	112	212	emit
X103	1	2	113	213	emit
X104	1	2	114	214	emit
X105	1	2	115	215	emit
X106	1	2	116	216	emit
X107	1	2	117	217	emit
X108	1	2	118	218	emit
X109	1	2	119	219	emit

X110	1	2	120	220	emit
X111	1	2	121	221	emit
X112	1	2	122	222	emit
X113	1	2	123	223	emit
X114	1	2	124	224	emit
X115	1	2	125	225	emit
X116	1	2	126	226	emit
X117	1	2	127	227	emit
X118	1	2	128	228	emit
X119	1	2	129	229	emit
X120	1	2	130	230	emit
X121	1	2	131	231	emit
X122	1	2	132	232	emit
X123	1	2	133	233	emit
X124	1	2	134	234	emit
X125	1	2	135	235	emit
X126	1	2	136	236	emit
X127	1	2	137	237	emit
X128	1	2	138	238	emit
X129	1	2	139	239	emit
X130	1	2	140	240	emit
X131	1	2	141	241	emit
X132	1	2	142	242	emit
X133	1	2	143	243	emit
X134	1	2	144	244	emit
X135	1	2	145	245	emit
X136	1	2	146	246	emit
X137	1	2	147	247	emit
X138	1	2	148	248	emit

***Op Amp**

x200 7 8 301 0 306 opamp

***Comparators**

X301	500	501	405	511	601	comptr
X302	500	501	405	512	602	comptr
X303	500	501	405	513	603	comptr
X304	500	501	405	514	604	comptr
X305	500	501	405	515	605	comptr
X306	500	501	405	516	606	comptr

***transgate**

X401	511	1	2	503	502	701	transgate
X402	512	1	2	503	502	702	transgate
X403	513	1	2	503	502	703	transgate
X404	514	1	2	503	502	704	transgate
X405	515	1	2	503	502	705	transgate
X406	516	1	2	503	502	706	transgate

```

*****Decoder (Comparator Level -> Thermometer
Code)***** X501 500 701 702 703 704 705 706
801 802 803 804 805 806 decclthc
*****
*****
*****Decoder (Thermometer
Code)***** X601 500 801
802 803 804 805 806 901 902 903 decthcsnsb
*****
***** *Voltage Shifter
M100 405 306 7 1 npf L=2u W=9u
M101 405 405 8 2 nnf L=2u W=5u

```

```

*Folding Subcircuit*****
*          V+ V-  Vin  Vout M5 M6  Vref
.subckt fold1 1 20  4   30  7  9   11

```

```

*Current sources
I1  8 20  .2mA
I2  5 20  .2mA

```

```

*Branch Ammeters
VID1 7 40 0V
VID2 9 41 0V
VID3 3 42 0V
VID4 10 43 0V
*VID9 1 55 0V
*VID10 57 20 0V

```

```

*MOSFETS***
*Diffamps
M1  40 3 8 20 nnf L=2u W=130u
M2  41 10 8 20 nnf L=2u W=130u

M3  42 4 5 20 nnf L=2u W=164u
M4  43 11 5 20 nnf L=2u W=164u

```

```

*Active Loads
M7  3 3 1 1 npf L=3u W=5u
M8  10 10 1 1 npf L=3u W=5u

```

```

*Source Follower/Voltage Shifter
M9  30 8 1 1 npf L=2u W=140u
M10 30 30 20 20 nnf L=2u W=5u
.ends

```

```

*Folding Subcircuit*****
*          V+ V-  Vin  Vout M5 M6  Vref

```

```
.subckt fold2 1 20 4 30 7 9 11
```

```
*Current sources
```

```
I1 8 20 .2mA
```

```
I2 5 20 .2mA
```

```
*Branch Ampermeters
```

```
VID1 7 40 0V
```

```
VID2 9 41 0V
```

```
VID3 3 42 0V
```

```
VID4 10 43 0V
```

```
*VID9 1 55 0V
```

```
*VID10 57 20 0V
```

```
*MOSFETS***
```

```
*Diffamps
```

```
M1 40 3 8 20 nnf L=2u W=130u
```

```
M2 41 10 8 20 nnf L=2u W=130u
```

```
M3 42 4 5 20 nnf L=2u W=168u
```

```
M4 43 11 5 20 nnf L=2u W=168u
```

```
*Active Loads
```

```
M7 3 3 1 1 npf L=3u W=5u
```

```
M8 10 10 1 1 npf L=3u W=5u
```

```
*Source Follower/Voltage Shifter
```

```
M9 30 8 1 1 npf L=2u W=140u
```

```
M10 30 30 20 20 nnf L=2u W=5u
```

```
.ends
```

```
*Folding Subcircuit*****
```

```
* V+ V- Vin Vout M5 M6 Vref
```

```
.subckt fold3 1 20 4 30 7 9 11
```

```
*Current sources
```

```
I1 8 20 .2mA
```

```
I2 5 20 .2mA
```

```
*Branch Ampermeters
```

```
VID1 7 40 0V
```

```
VID2 9 41 0V
```

```
VID3 3 42 0V
```

```
VID4 10 43 0V
```

```
*VID9 1 55 0V
```

```
*VID10 57 20 0V
```

```
*MOSFETS***
```

```
*Diffamps
```

```
M1 40 3 8 20 nnf L=2u W=130u
```

```
M2 41 10 8 20 nnf L=2u W=130u
```

```

M3  42 4 5 20  nnf L=2u  W=170u
M4  43 11 5 20 nnf  L=2u  W=170u

*Active Loads
M7  3 3 1 1  npf L=3u  W=5u
M8  10 10 1 1  npf L=3u  W=5u

*Source Follower/Voltage Shifter
M9  30 8 1 1 npf  L=2u  W=140u
M10 30 30 20 20 nnf L=2u W=5u
.ends

*Folding Subcircuit*****
*          V+ V-  Vin  Vout M5 M6  Vref
.subckt  fold4 1 20  4    30 7 9   11

*Current sources
I1  8 20    .2mA
I2  5 20    .2mA

*Branch Ampmeters
VID1 7 40 0V
VID2 9 41 0V
VID3 3 42 0V
VID4 10 43 0V
*VID9 1 55 0V
*VID10 57 20 0V
*MOSFETS***
*Diffamps
M1  40 3 8 20 nnf L=2u  W=130u
M2  41 10 8 20 nnf L=2u  W=130u

M3  42 4 5 20  nnf L=2u  W=172u
M4  43 11 5 20 nnf  L=2u  W=172u

*Active Loads
M7  3 3 1 1  npf L=3u  W=5u
M8  10 10 1 1  npf L=3u  W=5u

*Source Follower/Voltage Shifter
M9  30 8 1 1 npf  L=2u  W=140u
M10 30 30 20 20 nnf L=2u W=5u
.ends

*Folding Subcircuit*****
*          V+ V-  Vin  Vout M5 M6  Vref
.subckt  fold5 1 20  4    30 7 9   11

*Current sources

```

I1 8 20 .2mA
I2 5 20 .2mA

*Branch Ammeters

VID1 7 40 0V

VID2 9 41 0V

VID3 3 42 0V

VID4 10 43 0V

*VID9 1 55 0V

*VID10 57 20 0V

*MOSFETS***

*Diffamps

M1 40 3 8 20 nnf L=2u W=130u

M2 41 10 8 20 nnf L=2u W=130u

M3 42 4 5 20 nnf L=2u W=174u

M4 43 11 5 20 nnf L=2u W=174u

*Active Loads

M7 3 3 1 1 npf L=3u W=5u

M8 10 10 1 1 npf L=3u W=5u

*Source Follower/Voltage Shifter

M9 30 8 1 1 npf L=2u W=140u

M10 30 30 20 20 nnf L=2u W=5u

.ends

*Folding Subcircuit*****

* V+ V- Vin Vout M5 M6 Vref
.subckt fold6 1 20 4 30 7 9 11

*Current sources

I1 8 20 .2mA

I2 5 20 .2mA

*Branch Ammeters

VID1 7 40 0V

VID2 9 41 0V

VID3 3 42 0V

VID4 10 43 0V

*VID9 1 55 0V

*VID10 57 20 0V

*MOSFETS***

*Diffamps

M1 40 3 8 20 nnf L=2u W=130u

M2 41 10 8 20 nnf L=2u W=130u

M3 42 4 5 20 nnf L=2u W=176u

M4 43 11 5 20 nnf L=2u W=176u

*Active Loads

M7 3 3 1 1 npf L=3u W=5u
M8 10 10 1 1 npf L=3u W=5u

*Source Follower/Voltage Shifter

M9 30 8 1 1 npf L=2u W=140u
M10 30 30 20 20 nnf L=2u W=5u
.ends

*Folding Subcircuit*****

* V+ V- Vin Vout M5 M6 Vref
.subckt fold7 1 20 4 30 7 9 11

*Current sources

I1 8 20 .2mA
I2 5 20 .2mA

*Branch Ammeters

VID1 7 40 0V
VID2 9 41 0V
VID3 3 42 0V
VID4 10 43 0V
*VID9 1 55 0V
*VID10 57 20 0V

*MOSFETS***

*Diffamps

M1 40 3 8 20 nnf L=2u W=130u
M2 41 10 8 20 nnf L=2u W=130u

M3 42 4 5 20 nnf L=2u W=178u
M4 43 11 5 20 nnf L=2u W=178u

*Active Loads

M7 3 3 1 1 npf L=3u W=5u
M8 10 10 1 1 npf L=3u W=5u

*Source Follower/Voltage Shifter

M9 30 8 1 1 npf L=2u W=140u
M10 30 30 20 20 nnf L=2u W=5u
.ends

*Folding Subcircuit*****

* V+ V- Vin Vout M5 M6 Vref
.subckt fold8 1 20 4 30 7 9 11

*Current sources

I1 8 20 .2mA
I2 5 20 .2mA

*Branch Ampmeters

VID1 7 40 0V

VID2 9 41 0V

VID3 3 42 0V

VID4 10 43 0V

*VID9 1 55 0V

*VID10 57 20 0V

*MOSFETS***

*Diffamps

M1 40 3 8 20 nnf L=2u W=130u

M2 41 10 8 20 nnf L=2u W=130u

M3 42 4 5 20 nnf L=2u W=180u

M4 43 11 5 20 nnf L=2u W=180u

*Active Loads

M7 3 3 1 1 npf L=3u W=5u

M8 10 10 1 1 npf L=3u W=5u

*Source Follower/Voltage Shifter

M9 30 8 1 1 npf L=2u W=140u

M10 30 30 20 20 nnf L=2u W=5u

.ends

*Folding Subcircuit*****

	V+	V-	Vin	Vout	M5	M6	Vref
.subckt fold9	1	20	4	30	7	9	11

*Current sources

I1 8 20 .2mA

I2 5 20 .2mA

*Branch Ampmeters

VID1 7 40 0V

VID2 9 41 0V

VID3 3 42 0V

VID4 10 43 0V

*VID9 1 55 0V

*VID10 57 20 0V

*MOSFETS***

*Diffamps

M1 40 3 8 20 nnf L=2u W=130u

M2 41 10 8 20 nnf L=2u W=130u

M3 42 4 5 20 nnf L=2u W=180u

M4 43 11 5 20 nnf L=2u W=180u

*Active Loads

```
M7  3 3 1 1  npf L=3u  W=5u
M8  10 10 1 1  npf L=3u  W=5u
```

```
*Source Follower/Voltage Shifter
```

```
M9  30 8 1 1  npf  L=2u  W=140u
M10 30 30 20 20 nnf L=2u W=5u
.ends
```

```
*Folding Subcircuit*****
```

```
*      V+ V-  Vin  Vout M5 M6  Vref
.subckt fold10 1 20  4    30 7 9  11
```

```
*Current sources
```

```
I1  8 20  .2mA
I2  5 20  .2mA
```

```
*Branch Ammeters
```

```
VID1 7 40 0V
VID2 9 41 0V
VID3 3 42 0V
VID4 10 43 0V
*VID9 1 55 0V
*VID10 57 20 0V
```

```
*MOSFETS***
```

```
*Diffamps
```

```
M1  40 3 8 20 nnf L=2u  W=130u
M2  41 10 8 20 nnf L=2u  W=130u
```

```
M3  42 4 5 20  nnf L=2u  W=186u
M4  43 11 5 20 nnf  L=2u  W=186u
```

```
*Active Loads
```

```
M7  3 3 1 1  npf L=3u  W=5u
M8  10 10 1 1  npf L=3u  W=5u
```

```
*Source Follower/Voltage Shifter
```

```
M9  30 8 1 1  npf  L=2u  W=140u
M10 30 30 20 20 nnf L=2u W=5u
.ends
```

```
*Folding Subcircuit*****
```

```
*      V+ V-  Vin  Vout M5 M6  Vref
.subckt fold11 1 20  4    30 7 9  11
```

```
*Current sources
```

```
I1  8 20  .2mA
I2  5 20  .2mA
```

```

*Branch Ammeters
VID1 7 40 0V
VID2 9 41 0V
VID3 3 42 0V
VID4 10 43 0V
*VID9 1 55 0V
*VID10 57 20 0V
*MOSFETS***
*Diffamps
M1 40 3 8 20 nnf L=2u W=130u
M2 41 10 8 20 nnf L=2u W=130u

M3 42 4 5 20 nnf L=2u W=188u
M4 43 11 5 20 nnf L=2u W=188u

*Active Loads
M7 3 3 1 1 npf L=3u W=5u
M8 10 10 1 1 npf L=3u W=5u

*Source Follower/Voltage Shifter
M9 30 8 1 1 npf L=2u W=140u
M10 30 30 20 20 nnf L=2u W=5u
.ends

*Folding Subcircuit*****
* V+ V- Vin Vout M5 M6 Vref
.subckt fold12 1 20 4 30 7 9 11

*Current sources
I1 8 20 .2mA
I2 5 20 .2mA

*Branch Ammeters
VID1 7 40 0V
VID2 9 41 0V
VID3 3 42 0V
VID4 10 43 0V
*VID9 1 55 0V
*VID10 57 20 0V
*MOSFETS***
*Diffamps
M1 40 3 8 20 nnf L=2u W=130u
M2 41 10 8 20 nnf L=2u W=130u

M3 42 4 5 20 nnf L=2u W=192u
M4 43 11 5 20 nnf L=2u W=192u

*Active Loads
M7 3 3 1 1 npf L=3u W=5u

```

M8 10 10 1 1 npf L=3u W=5u

*Source Follower/Voltage Shifter

M9 30 8 1 1 npf L=2u W=140u

M10 30 30 20 20 nnf L=2u W=5u

.ends

*Folding Subcircuit*****

* V+ V- Vin Vout M5 M6 Vref
.subckt fold13 1 20 4 30 7 9 11

*Current sources

I1 8 20 .2mA

I2 5 20 .2mA

*Branch Ammeters

VID1 7 40 0V

VID2 9 41 0V

VID3 3 42 0V

VID4 10 43 0V

*VID9 1 55 0V

*VID10 57 20 0V

*MOSFETS***

*Diffamps

M1 40 3 8 20 nnf L=2u W=130u

M2 41 10 8 20 nnf L=2u W=130u

M3 42 4 5 20 nnf L=2u W=196u

M4 43 11 5 20 nnf L=2u W=196u

*Active Loads

M7 3 3 1 1 npf L=3u W=5u

M8 10 10 1 1 npf L=3u W=5u

*Source Follower/Voltage Shifter

M9 30 8 1 1 npf L=2u W=140u

M10 30 30 20 20 nnf L=2u W=5u

.ends

*Folding Subcircuit*****

* V+ V- Vin Vout M5 M6 Vref
.subckt fold14 1 20 4 30 7 9 11

*Current sources

I1 8 20 .2mA

I2 5 20 .2mA

```

*Branch Ampermeters
VID1 7 40 0V
VID2 9 41 0V
VID3 3 42 0V
VID4 10 43 0V
*VID9 1 55 0V
*VID10 57 20 0V
*MOSFETS***
*Diffamps
M1 40 3 8 20 nnf L=2u W=130u
M2 41 10 8 20 nnf L=2u W=130u

M3 42 4 5 20 nnf L=2u W=200u
M4 43 11 5 20 nnf L=2u W=200u

*Active Loads
M7 3 3 1 1 npf L=3u W=5u
M8 10 10 1 1 npf L=3u W=5u

*Source Follower/Voltage Shifter
M9 30 8 1 1 npf L=2u W=140u
M10 30 30 20 20 nnf L=2u W=5u
.ends

*Folding Subcircuit*****
* V+ V- Vin Vout M5 M6 Vref
.subckt fold15 1 20 4 30 7 9 11

*Current sources
I1 8 20 .2mA
I2 5 20 .2mA

*Branch Ampermeters
VID1 7 40 0V
VID2 9 41 0V
VID3 3 42 0V
VID4 10 43 0V
*VID9 1 55 0V
*VID10 57 20 0V
*MOSFETS***
*Diffamps
M1 40 3 8 20 nnf L=2u W=130u
M2 41 10 8 20 nnf L=2u W=130u

M3 42 4 5 20 nnf L=2u W=204u
M4 43 11 5 20 nnf L=2u W=204u

*Active Loads
M7 3 3 1 1 npf L=3u W=5u

```

M8 10 10 1 1 npf L=3u W=5u

*Source Follower/Voltage Shifter

M9 30 8 1 1 npf L=2u W=140u

M10 30 30 20 20 nnf L=2u W=5u

.ends

*Folding Subcircuit*****

* V+ V- Vin Vout M5 M6 Vref

.subckt fold16 1 20 4 30 7 9 11

*Current sources

I1 8 20 .2mA

I2 5 20 .2mA

*Branch Ammeters

VID1 7 40 0V

VID2 9 41 0V

VID3 3 42 0V

VID4 10 43 0V

*VID9 1 55 0V

*VID10 57 20 0V

*MOSFETS***

*Diffamps

M1 40 3 8 20 nnf L=2u W=130u

M2 41 10 8 20 nnf L=2u W=130u

M3 42 4 5 20 nnf L=2u W=208u

M4 43 11 5 20 nnf L=2u W=208u

*Active Loads

M7 3 3 1 1 npf L=3u W=5u

M8 10 10 1 1 npf L=3u W=5u

*Source Follower/Voltage Shifter

M9 30 8 1 1 npf L=2u W=140u

M10 30 30 20 20 nnf L=2u W=5u

.ends

*Folding Subcircuit*****

* V+ V- Vin Vout M5 M6 Vref

.subckt fold17 1 20 4 30 7 9 11

*Current sources

I1 8 20 .2mA

I2 5 20 .2mA

***Branch Ampmeters**

VID1 7 40 0V

VID2 9 41 0V

VID3 3 42 0V

VID4 10 43 0V

*VID9 1 55 0V

*VID10 57 20 0V

MOSFETS**

***Diffamps**

M1 40 3 8 20 nnf L=2u W=130u

M2 41 10 8 20 nnf L=2u W=130u

M3 42 4 5 20 nnf L=2u W=212u

M4 43 11 5 20 nnf L=2u W=212u

***Active Loads**

M7 3 3 1 1 npf L=3u W=5u

M8 10 10 1 1 npf L=3u W=5u

***Source Follower/Voltage Shifter**

M9 30 8 1 1 npf L=2u W=140u

M10 30 30 20 20 nnf L=2u W=5u

.ends

Folding Subcircuit****

* V+ V- Vin Vout M5 M6 Vref
.subckt fold18 1 20 4 30 7 9 11

***Current sources**

I1 8 20 .2mA

I2 5 20 .2mA

***Branch Ampmeters**

VID1 7 40 0V

VID2 9 41 0V

VID3 3 42 0V

VID4 10 43 0V

*VID9 1 55 0V

*VID10 57 20 0V

MOSFETS**

***Diffamps**

M1 40 3 8 20 nnf L=2u W=130u

M2 41 10 8 20 nnf L=2u W=130u

M3 42 4 5 20 nnf L=2u W=216u

M4 43 11 5 20 nnf L=2u W=216u

***Active Loads**

M7 3 3 1 1 npf L=3u W=5u

M8 10 10 1 1 npf L=3u W=5u

*Source Follower/Voltage Shifter

M9 30 8 1 1 npf L=2u W=140u

M10 30 30 20 20 nnf L=2u W=5u

.ends

*Folding Subcircuit*****

```
*      V+ V-  Vin  Vout M5 M6  Vref
.subckt fold19 1 20  4   30 7 9  11
```

*Current sources

I1 8 20 .2mA

I2 5 20 .2mA

*Branch Ammeters

VID1 7 40 0V

VID2 9 41 0V

VID3 3 42 0V

VID4 10 43 0V

*VID9 1 55 0V

*VID10 57 20 0V

*MOSFETS***

*Diffamps

M1 40 3 8 20 nnf L=2u W=130u

M2 41 10 8 20 nnf L=2u W=130u

M3 42 4 5 20 nnf L=2u W=220u

M4 43 11 5 20 nnf L=2u W=220u

*Active Loads

M7 3 3 1 1 npf L=3u W=5u

M8 10 10 1 1 npf L=3u W=5u

*Source Follower/Voltage Shifter

M9 30 8 1 1 npf L=2u W=140u

M10 30 30 20 20 nnf L=2u W=5u

.ends

*Folding Subcircuit*****

```
*      V+ V-  Vin  Vout M5 M6  Vref
.subckt fold20 1 20  4   30 7 9  11
```

*Current sources

I1 8 20 .2mA

I2 5 20 .2mA

***Branch Ampermeters**

VID1 7 40 0V

VID2 9 41 0V

VID3 3 42 0V

VID4 10 43 0V

*VID9 1 55 0V

*VID10 57 20 0V

MOSFETS**

***Diffamps**

M1 40 3 8 20 nnf L=2u W=130u

M2 41 10 8 20 nnf L=2u W=130u

M3 42 4 5 20 nnf L=2u W=224u

M4 43 11 5 20 nnf L=2u W=224u

***Active Loads**

M7 3 3 1 1 npf L=3u W=5u

M8 10 10 1 1 npf L=3u W=5u

***Source Follower/Voltage Shifter**

M9 30 8 1 1 npf L=2u W=140u

M10 30 30 20 20 nnf L=2u W=5u

.ends

Folding Subcircuit****

* V+ V- Vin Vout M5 M6 Vref
.subckt fold21 1 20 4 30 7 9 11

***Current sources**

I1 8 20 .2mA

I2 5 20 .2mA

***Branch Ampermeters**

VID1 7 40 0V

VID2 9 41 0V

VID3 3 42 0V

VID4 10 43 0V

*VID9 1 55 0V

*VID10 57 20 0V

MOSFETS**

***Diffamps**

M1 40 3 8 20 nnf L=2u W=130u

M2 41 10 8 20 nnf L=2u W=130u

M3 42 4 5 20 nnf L=2u W=228u

M4 43 11 5 20 nnf L=2u W=228u

***Active Loads**

M7 3 3 1 1 npf L=3u W=5u

M8 10 10 1 1 npf L=3u W=5u

```

*Source Follower/Voltage Shifter
M9  30 8 1 1 npf L=2u W=140u
M10 30 30 20 20 nnf L=2u W=5u
.ends

```

```

*Folding Subcircuit*****
*          V+ V-  Vin  Vout M5 M6  Vref
.subckt  fold22  1  20   4    30  7  9   11

```

```

*Current sources
I1   8 20   .2mA
I2   5 20   .2mA

```

```

*Branch Ammeters

```

```

VID1 7 40 0V
VID2 9 41 0V
VID3 3 42 0V
VID4 10 43 0V
*VID9 1 55 0V
*VID10 57 20 0V

```

```

*MOSFETS***

```

```

*Diffamps

```

```

M1    40 3 8 20 nnf L=2u  W=130u
M2    41 10 8 20 nnf L=2u  W=130u

```

```

M3    42 4 5 20  nnf L=2u  W=232u
M4    43 11 5 20 nnf  L=2u  W=232u

```

```

*Active Loads

```

```

M7   3 3 1 1  npf L=3u  W=5u
M8   10 10 1 1  npf L=3u  W=5u

```

```

*Source Follower/Voltage Shifter

```

```

M9  30 8 1 1 npf L=2u W=140u
M10 30 30 20 20 nnf L=2u W=5u
.ends

```

```

*Folding Subcircuit*****
*          V+ V-  Vin  Vout M5 M6  Vref
.subckt  fold23  1  20   4    30  7  9   11

```

```

*Current sources
I1   8 20   .2mA
I2   5 20   .2mA

```

```

*Branch Ammeters

```

```

VID1 7 40 0V
VID2 9 41 0V
VID3 3 42 0V

```

```

VID4 10 43 0V
*VID9 1 55 0V
*VID10 57 20 0V
*MOSFETS***
*Diffamps
M1 40 3 8 20 nnf L=2u W=130u
M2 41 10 8 20 nnf L=2u W=130u

M3 42 4 5 20 nnf L=2u W=236u
M4 43 11 5 20 nnf L=2u W=236u

*Active Loads
M7 3 3 1 1 npf L=3u W=5u
M8 10 10 1 1 npf L=3u W=5u

*Source Follower/Voltage Shifter
M9 30 8 1 1 npf L=2u W=140u
M10 30 30 20 20 nnf L=2u W=5u
.ends

*Folding Subcircuit*****
* V+ V- Vin Vout M5 M6 Vref
.subckt fold24 1 20 4 30 7 9 11

*Current sources
I1 8 20 .2mA
I2 5 20 .2mA

*Branch Ampmeters
VID1 7 40 0V
VID2 9 41 0V
VID3 3 42 0V
VID4 10 43 0V
*VID9 1 55 0V
*VID10 57 20 0V
*MOSFETS***
*Diffamps
M1 40 3 8 20 nnf L=2u W=130u
M2 41 10 8 20 nnf L=2u W=130u

M3 42 4 5 20 nnf L=2u W=240u
M4 43 11 5 20 nnf L=2u W=240u

*Active Loads
M7 3 3 1 1 npf L=3u W=5u
M8 10 10 1 1 npf L=3u W=5u

*Source Follower/Voltage Shifter
M9 30 8 1 1 npf L=2u W=140u
M10 30 30 20 20 nnf L=2u W=5u

```

.ends

*Folding Subcircuit*****

```
*          V+ V-  Vin  Vout M5 M6  Vref
.subckt  fold25  1  20   4    30  7  9   11
```

*Current sources

```
I1   8 20   .2mA
I2   5 20   .2mA
```

*Branch Ammeters

```
VID1 7 40 0V
VID2 9 41 0V
VID3 3 42 0V
VID4 10 43 0V
*VID9 1 55 0V
*VID10 57 20 0V
```

*MOSFETS***

*Diffamps

```
M1    40 3 8 20 nnf L=2u  W=130u
M2    41 10 8 20 nnf L=2u  W=130u
```

```
M3    42 4 5 20 nnf L=2u  W=244u
M4    43 11 5 20 nnf L=2u  W=244u
```

*Active Loads

```
M7    3 3 1 1 npf L=3u  W=5u
M8    10 10 1 1 npf L=3u  W=5u
```

*Source Follower/Voltage Shifter

```
M9    30 8 1 1 npf L=2u  W=140u
M10   30 30 20 20 nnf L=2u W=5u
```

.ends

*Folding Subcircuit*****

```
*          V+ V-  Vin  Vout M5 M6  Vref
.subckt  fold26  1  20   4    30  7  9   11
```

*Current sources

```
I1   8 20   .2mA
I2   5 20   .2mA
```

*Branch Ammeters

```
VID1 7 40 0V
VID2 9 41 0V
VID3 3 42 0V
VID4 10 43 0V
*VID9 1 55 0V
*VID10 57 20 0V
```

```

*MOSFETS***
*Diffamps
M1      40 3 8 20 nnf L=2u W=130u
M2      41 10 8 20 nnf L=2u W=130u

M3      42 4 5 20 nnf L=2u W=248u
M4      43 11 5 20 nnf L=2u W=248u

*Active Loads
M7      3 3 1 1 npf L=3u W=5u
M8      10 10 1 1 npf L=3u W=5u

*Source Follower/Voltage Shifter
M9      30 8 1 1 npf L=2u W=140u
M10     30 30 20 20 nnf L=2u W=5u
.ends

*Folding Subcircuit*****
*          V+ V-  Vin  Vout M5 M6  Vref
.subckt fold27 1 20 4 30 7 9 11

*Current sources
I1      8 20 .2mA
I2      5 20 .2mA

*Branch Ampmeters
VID1 7 40 0V
VID2 9 41 0V
VID3 3 42 0V
VID4 10 43 0V
*VID9 1 55 0V
*VID10 57 20 0V
*MOSFETS***
*Diffamps
M1      40 3 8 20 nnf L=2u W=130u
M2      41 10 8 20 nnf L=2u W=130u

M3      42 4 5 20 nnf L=2u W=252u
M4      43 11 5 20 nnf L=2u W=252u

*Active Loads
M7      3 3 1 1 npf L=3u W=5u
M8      10 10 1 1 npf L=3u W=5u

*Source Follower/Voltage Shifter
M9      30 8 1 1 npf L=2u W=140u
M10     30 30 20 20 nnf L=2u W=5u
.ends

*Folding Subcircuit*****

```

```

*
      V+ V-  Vin  Vout M5 M6  Vref
.subckt fold28 1  20  4    30 7  9   11

```

*Current sources

```

I1   8 20   .2mA
I2   5 20   .2mA

```

*Branch Ampmeters

```

VID1 7 40 0V
VID2 9 41 0V
VID3 3 42 0V
VID4 10 43 0V
*VID9 1 55 0V
*VID10 57 20 0V

```

*MOSFETS***

*Diffamps

```

M1    40 3 8 20 nnf L=2u  W=130u
M2    41 10 8 20 nnf L=2u  W=130u

```

```

M3    42 4 5 20 nnf L=2u  W=258u
M4    43 11 5 20 nnf L=2u  W=258u

```

*Active Loads

```

M7    3 3 1 1 npf L=3u  W=5u
M8    10 10 1 1 npf L=3u  W=5u

```

*Source Follower/Voltage Shifter

```

M9    30 8 1 1 npf L=2u  W=140u
M10   30 30 20 20 nnf L=2u W=5u
.ends

```

*Folding Subcircuit*****

```

*
      V+ V-  Vin  Vout M5 M6  Vref
.subckt fold29 1  20  4    30 7  9   11

```

*Current sources

```

I1   8 20   .2mA
I2   5 20   .2mA

```

*Branch Ampmeters

```

VID1 7 40 0V
VID2 9 41 0V
VID3 3 42 0V
VID4 10 43 0V
*VID9 1 55 0V
*VID10 57 20 0V

```

*MOSFETS***

*Diffamps

```

M1    40 3 8 20 nnf L=2u  W=130u

```

M2 41 10 8 20 nnf L=2u W=130u

M3 42 4 5 20 nnf L=2u W=264u

M4 43 11 5 20 nnf L=2u W=264u

*Active Loads

M7 3 3 1 1 npf L=3u W=5u

M8 10 10 1 1 npf L=3u W=5u

*Source Follower/Voltage Shifter

M9 30 8 1 1 npf L=2u W=140u

M10 30 30 20 20 nnf L=2u W=5u

.ends

*Folding Subcircuit*****

* V+ V- Vin Vout M5 M6 Vref
.subckt fold30 1 20 4 30 7 9 11

*Current sources

I1 8 20 .2mA

I2 5 20 .2mA

*Branch Ammeters

VID1 7 40 0V

VID2 9 41 0V

VID3 3 42 0V

VID4 10 43 0V

*VID9 1 55 0V

*VID10 57 20 0V

*MOSFETS***

*Diffamps

M1 40 3 8 20 nnf L=2u W=130u

M2 41 10 8 20 nnf L=2u W=130u

M3 42 4 5 20 nnf L=2u W=270u

M4 43 11 5 20 nnf L=2u W=270u

*Active Loads

M7 3 3 1 1 npf L=3u W=5u

M8 10 10 1 1 npf L=3u W=5u

*Source Follower/Voltage Shifter

M9 30 8 1 1 npf L=2u W=140u

M10 30 30 20 20 nnf L=2u W=5u

.ends

*Folding Subcircuit*****

* V+ V- Vin Vout M5 M6 Vref
.subckt fold31 1 20 4 30 7 9 11

***Current sources**

I1 8 20 .2mA
I2 5 20 .2mA

***Branch Ampmeters**

VID1 7 40 0V
VID2 9 41 0V
VID3 3 42 0V
VID4 10 43 0V
*VID9 1 55 0V
*VID10 57 20 0V

MOSFETS**

***Diffamps**

M1 40 3 8 20 nnf L=2u W=130u
M2 41 10 8 20 nnf L=2u W=130u

M3 42 4 5 20 nnf L=2u W=282u
M4 43 11 5 20 nnf L=2u W=282u

***Active Loads**

M7 3 3 1 1 npf L=3u W=5u
M8 10 10 1 1 npf L=3u W=5u

***Source Follower/Voltage Shifter**

M9 30 8 1 1 npf L=2u W=140u
M10 30 30 20 20 nnf L=2u W=5u
.ends

Folding Subcircuit****

* V+ V- Vin Vout M5 M6 Vref
.subckt fold32 1 20 4 30 7 9 11

***Current sources**

I1 8 20 .2mA
I2 5 20 .2mA

***Branch Ampmeters**

VID1 7 40 0V
VID2 9 41 0V
VID3 3 42 0V
VID4 10 43 0V
*VID9 1 55 0V
*VID10 57 20 0V

MOSFETS**

***Diffamps**

M1 40 3 8 20 nnf L=2u W=130u
M2 41 10 8 20 nnf L=2u W=130u

```

M3  42 4 5 20  nnf L=2u  W=294u
M4  43 11 5 20  nnf  L=2u  W=294u

```

*Active Loads

```

M7  3 3 1 1  npf L=3u  W=5u
M8  10 10 1 1  npf L=3u  W=5u

```

*Source Follower/Voltage Shifter

```

M9  30 8 1 1  npf  L=2u  W=140u
M10 30 30 20 20 nnf L=2u W=5u
.ends

```

*Folding Subcircuit*****

```

*          V+ V-  Vin  Vout M5 M6  Vref
.subckt fold33 1 20  4    30 7 9  11

```

*Current sources

```

I1  8 20  .2mA
I2  5 20  .2mA

```

*Branch Ammeters

```

VID1 7 40 0V
VID2 9 41 0V
VID3 3 42 0V
VID4 10 43 0V
*VID9 1 55 0V
*VID10 57 20 0V
*MOSFETS***

```

*Diffamps

```

M1  40 3 8 20  nnf L=2u  W=130u
M2  41 10 8 20  nnf L=2u  W=130u

```

```

M3  42 4 5 20  nnf L=2u  W=296u
M4  43 11 5 20  nnf  L=2u  W=296u

```

*Active Loads

```

M7  3 3 1 1  npf L=3u  W=5u
M8  10 10 1 1  npf L=3u  W=5u

```

*Source Follower/Voltage Shifter

```

M9  30 8 1 1  npf  L=2u  W=140u
M10 30 30 20 20 nnf L=2u W=5u
.ends

```

*Folding Subcircuit*****

```

*          V+ V-  Vin  Vout M5 M6  Vref
.subckt fold34 1 20  4    30 7 9  11

```

***Current sources**

I1 8 20 .2mA

I2 5 20 .2mA

***Branch Ampmeters**

VID1 7 40 0V

VID2 9 41 0V

VID3 3 42 0V

VID4 10 43 0V

*VID9 1 55 0V

*VID10 57 20 0V

MOSFETS**

***Diffamps**

M1 40 3 8 20 nnf L=2u W=130u

M2 41 10 8 20 nnf L=2u W=130u

M3 42 4 5 20 nnf L=2u W=300u

M4 43 11 5 20 nnf L=2u W=300u

***Active Loads**

M7 3 3 1 1 npf L=3u W=5u

M8 10 10 1 1 npf L=3u W=5u

***Source Follower/Voltage Shifter**

M9 30 8 1 1 npf L=2u W=140u

M10 30 30 20 20 nnf L=2u W=5u

.ends

Folding Subcircuit****

* V+ V- Vin Vout M5 M6 Vref

.subckt fold35 1 20 4 30 7 9 11

***Current sources**

I1 8 20 .2mA

I2 5 20 .2mA

***Branch Ampmeters**

VID1 7 40 0V

VID2 9 41 0V

VID3 3 42 0V

VID4 10 43 0V

*VID9 1 55 0V

*VID10 57 20 0V

MOSFETS**

***Diffamps**

M1 40 3 8 20 nnf L=2u W=130u

M2 41 10 8 20 nnf L=2u W=130u

```
M3  42 4 5 20  nnf L=2u  W=310u
M4  43 11 5 20  nnf  L=2u  W=310u
```

***Active Loads**

```
M7   3 3 1 1  npf L=3u  W=5u
M8  10 10 1 1  npf L=3u  W=5u
```

***Source Follower/Voltage Shifter**

```
M9   30 8 1 1  npf  L=2u  W=140u
M10  30 30 20 20 nnf L=2u W=5u
.ends
```

Folding Subcircuit****

```
*          V+ V-  Vin  Vout M5 M6  Vref
.subckt fold36 1 20  4   30 7 9  11
```

***Current sources**

```
I1   8 20   .2mA
I2   5 20   .2mA
```

***Branch Ampermeters**

```
VID1 7 40 0V
VID2 9 41 0V
VID3 3 42 0V
VID4 10 43 0V
*VID9 1 55 0V
*VID10 57 20 0V
```

MOSFETS**

***Diffamps**

```
M1   40 3 8 20  nnf L=2u  W=130u
M2   41 10 8 20  nnf L=2u  W=130u
```

```
M3   42 4 5 20  nnf L=2u  W=320u
M4   43 11 5 20  nnf  L=2u  W=320u
```

***Active Loads**

```
M7   3 3 1 1  npf L=3u  W=5u
M8  10 10 1 1  npf L=3u  W=5u
```

***Source Follower/Voltage Shifter**

```
M9   30 8 1 1  npf  L=2u  W=140u
M10  30 30 20 20 nnf L=2u W=5u
.ends
```

Folding Subcircuit****

```
*          V+ V-  Vin  Vout M5 M6  Vref
.subckt fold37 1 20  4   30 7 9  11
```

***Current sources**

I1 8 20 .2mA
I2 5 20 .2mA

***Branch Ampmeters**

VID1 7 40 0V
VID2 9 41 0V
VID3 3 42 0V
VID4 10 43 0V
*VID9 1 55 0V
*VID10 57 20 0V

MOSFETS**

***Diffamps**

M1 40 3 8 20 nnf L=2u W=130u
M2 41 10 8 20 nnf L=2u W=130u

M3 42 4 5 20 nnf L=2u W=326u
M4 43 11 5 20 nnf L=2u W=326u

***Active Loads**

M7 3 3 1 1 npf L=3u W=5u
M8 10 10 1 1 npf L=3u W=5u

***Source Follower/Voltage Shifter**

M9 30 8 1 1 npf L=2u W=140u
M10 30 30 20 20 nnf L=2u W=5u
.ends

Folding Subcircuit****

* V+ V- Vin Vout M5 M6 Vref
.subckt fold38 1 20 4 30 7 9 11

***Current sources**

I1 8 20 .2mA
I2 5 20 .2mA

***Branch Ampmeters**

VID1 7 40 0V
VID2 9 41 0V
VID3 3 42 0V
VID4 10 43 0V
*VID9 1 55 0V
*VID10 57 20 0V

MOSFETS**

***Diffamps**

M1 40 3 8 20 nnf L=2u W=130u
M2 41 10 8 20 nnf L=2u W=130u

```

M3  42 4 5 20  nnf L=2u W=330u
M4  43 11 5 20  nnf L=2u W=330u

```

***Active Loads**

```

M7  3 3 1 1  npf L=3u W=5u
M8  10 10 1 1  npf L=3u W=5u

```

***Source Follower/Voltage Shifter**

```

M9  30 8 1 1  npf L=2u W=140u
M10 30 30 20 20  nnf L=2u W=5u
.ends

```

******Emitter Follower*******

```

*          V+ V- Vin Vout
.subckt emit 1 4 2 6
M2 1 2 6 4  nnf L=10u W=3u
.ends

```

```

*opamp subcircuit Vdd Vss V- V+ Vout
.subckt opamp 1 2 7 9 6

```

***Passive Elements**

```

Rbias 1 3 2K
Rz     5 10 13K
CC      10 6 5pF

```

***Fets**

```

M1  4 7 8 2  nnf L=10u W=40u
M2  5 9 8 2  nnf L=10u W=40u
M3  4 4 1 1  npf L=22u W=10u
M4  5 4 1 1  npf L=22u W=10u
M5  6 5 1 1  npf L=10u W=45u
M6  6 3 2 2  nnf L=10u W=10u
M7  8 3 2 2  nnf L=50u W=10u
M8  3 3 2 2  nnf L=10u W=10u

```

.ends

*******Comparator Subcircuits*******

```

*          Vdd Vss Vn Vout Vp
.subckt comptr 1 2 6 9 8
VB 10 0 dc .8v
M1  4 6 7 0  nnf W=10u L=10u
M2  5 8 7 0  nnf W=10u L=10u
M3  4 4 1 1  npf W=12u L=3u
M4  5 4 1 1  npf W=12u L=3u

```

```

M5  7 10 2 0  nnf W=10u L=10u
M6  9 5 1 1  npf W=4u L=2u
M7  9 5 0 0  nnf W=4u L=2u

```

```

.ends

```

```

****Transmission Gate ****

```

```

*          Vin VDD VSS  VC1 VC2 Vout
.subckt transgate 1  10 20   4  2  3

```

```

*Vdd 10 0 5

```

```

*VCC 20 0 0.0

```

```

M1  3 2 1 10  npf  W=3u L=2u

```

```

M2  1 4 3 20  nnf  w=6u L=2u

```

```

C1 3 0 1nf ic=0

```

```

.ends

```

```

****Decoder (Comparator Level to Thermometer Code)****

```

```

*          Vdd v1 v2 v3 v4 v5 v6 vo1 vo2 vo3 vo4 vo5
vo6 .subckt decclthc 1  2 3  4  5  6  7  21 22 23 24
25 26

```

```

*Inverter Instantiations

```

```

X1  1 3 11 inv

```

```

X2  1 4 12 inv

```

```

X3  1 5 13 inv

```

```

X4  1 6 14 inv

```

```

X5  1 7 15 inv

```

```

X6  1 0 16 inv

```

```

*AND 2-input Instantiations

```

```

X101 1 2 11 21 and2

```

```

X102 1 3 12 22 and2

```

```

X103 1 4 13 23 and2

```

```

X104 1 5 14 24 and2

```

```

X105 1 6 15 25 and2

```

```

X106 1 7 16 26 and2

```

```

*****Subcircuit Definitions*****

```

```

*****Inverter Subcircuit*****

```

```

*          Vdd  Vin  Vout
.subckt inv  1    2    3

```

```

*Fet's

```

```

M1  3 2 1 1  npf  W=7u L=2u
M2  3 2 0 0  nnf  W=3u L=2u

```

```

.ends

```

```

*****AND 2-Input Subcircuit*****

```

```

*          Vdd  VA  VB  Vout
.subckt  and2    1   2   3    6

```

```

*FET's

```

```

M1  4 2 1 1  npf  W=7u L=2u
M2  4 3 1 1  npf  W=7u L=2u
M3  6 4 1 1  npf  W=7u L=2u
M4  4 3 5 0  nnf  W=3u L=2u
M5  5 2 0 0  nnf  W=3u L=2u
M6  6 4 0 0  nnf  W=3u L=2u

```

```

.ends

```

```

.ends

```

```

*****Decoder (Thermometer Code to SNS Binary)

```

```

*****

```

```

.subckt  decthcsnsb  1 2 3 4 5 6 7  101 102 103

```

```

*OR Gate Instantiations

```

```

X1  1 2 4 6 101  or3
X2  1 3 4 7 102  or3
X3  1 5 6 7 103  or3

```

```

****OR Gate Subcircuit****

```

```

*          Vdd  VA  VB  VC  Vout
.subckt  or3    1   3   5   7   11

```

```

*FET's

```

```

M1  4 3 1 1  npf  W=7u L=2u
M2  6 5 4 1  npf  W=7u L=2u
M3  10 7 6 1  npf  W=7u L=2u
M5  11 10 1 1  npf  W=7u L=2u
M6  10 3 0 0  nnf  W=3u L=2u
M7  10 5 0 0  nnf  W=3u L=2u
M8  10 7 0 0  nnf  W=3u L=2u
M10 11 10 0 0  nnf  W=3u L=2u

```

.ends

.ends

****Simulation Parameters****

.option dcon=1 post probe
*.dc vin 0.0V .82V .029286

*.probe tran v(3) v(405) V(701) V(702) V(703) V(704) V(705)
V(706) V(801) V(802) *V(803) V(804) V(805) V(806) V(901)
V(902) V(903) .probe v(405)

.tran 65.6ns 1.63us

.end

APPENDIX B
INPUT FILE FOR PLA GENERATION

```
#Thesis PLA
a pla
b  I10 I9 I8 I7 I6 I5 I4 I3 I2 I1
    X9 X8 X7 X6 X5 X4 X3 X2 X1
#.type fr
#####PLA Truth Table#####
10
9
512
0000000000 0000000000
0001001001 0000000001
0010010010 0000000010
0011011011 0000000011
0100100100 0000000100
0101101101 0000000101
0110110110 0000000110
0111111110 0000000111
1000111101 0000001000
1001110100 0000001001
1010101011 0000001010
1010100010 0000001011
1001011001 0000001100
1000010000 0000001101
0111001000 0000001110
0110000001 0000001111
0101000010 0000010000
0100001011 0000010001
0011010100 0000010010
0010011101 0000010011
0001100110 0000010100
0000101110 0000010101
0000110101 0000010110
0001111100 0000010111
0010111011 0000011000
0011110010 0000011001
0100101001 0000011010
0101100000 0000011011
0110011000 0000011100
0111010001 0000011101
1000001010 0000011110
1001000011 0000011111
1010000100 0001000000
1010001101 0001000001
1001010110 0001000010
1000011110 0001000011
```

0111100101	000100100
0110101100	000100101
0101110011	000100110
0100111010	000100111
0011111001	000101000
0010110000	000101001
0001101000	000101010
0000100001	000101011
0000011010	000101100
0001010011	000101101
0010001100	000101110
0011000101	000101111
0100000110	000110000
0101001110	000110001
0110010101	000110010
0111011100	000110011
1000100011	000110100
1001101010	000110101
1010110001	000110110
1010111000	000110111
1001111000	000111000
1000110001	000111001
0111101010	000111010
0110100011	000111011
0101011100	000111100
0100010101	000111101
0011001110	000111110
0010000110	000111111
0001000101	001000000
0000001100	001000001
0000010011	001000010
0001011010	001000011
0010100001	001000100
0011101000	001000101
0100110000	001000110
0101111001	001000111
0110111010	001001000
0111110011	001001001
1000101100	001001010
1001100101	001001011
1010011110	001001100
1010010110	001001101
1001001101	001001110
1000000100	001001111
0111000011	001010000
0110001010	001010001
0101010001	001010010
0100011000	001010011
0011100000	001010100
0010101001	001010101
0001110010	001010110

0000111011	001010111
0000111100	001011000
0001110101	001011001
0010101110	001011010
0011100110	001011011
0100011101	001011100
0101010100	001011101
0110001011	001011110
0111000010	001011111
1000000001	001100000
1001001000	001100001
1010010000	001100010
1010011001	001100011
1001100010	001100100
1000101011	001100101
0111110100	001100110
0110111101	001100111
0101111110	001101000
0100110110	001101001
0011101101	001101010
0010100100	001101011
0001011011	001101100
0000010010	001101101
0000001001	001101110
0001000000	001101111
0010000000	001110000
0011001001	001110001
0100010010	001110010
0101011011	001110011
0110100100	001110100
0111101101	001110101
1000110110	001110110
1001111110	001110111
1010111101	001111000
1010110100	001111001
1001101011	001111010
1000100010	001111011
0111011001	001111100
0110010000	001111101
0101001000	001111110
0100000001	001111111
0011000010	010000000
0010001011	010000001
0001010100	010000010
0000011101	010000011
0000100110	010000100
0001101110	010000101
0010110101	010000110
0011111100	010000111
0100111011	010001000
0101110010	010001001

0110101001	010001010
0111100000	010001011
1000011000	010001100
1001010001	010001101
1010001010	010001110
1010000011	010001111
1001000100	010010000
1000001101	010010001
0111010110	010010010
0110011110	010010011
0101100101	010010100
0100101100	010010101
0011110011	010010110
0010111010	010010111
0001111001	010011000
0000110000	010011001
0000101000	010011010
0001100001	010011011
0010011010	010011100
0011010011	010011101
0100001100	010011110
0101000101	010011111
0110000110	010100000
0111001110	010100001
1000010101	010100010
1001011100	010100011
1010100011	010100100
1010101010	010100101
1001110001	010100110
1000111000	010100111
0111111000	010101000
0110110001	010101001
0101101010	010101010
0100100011	010101011
0011011100	010101100
0010010101	010101101
0001001110	010101110
0000000110	010101111
0000000101	010110000
0001001100	010110001
0010010011	010110010
0011011010	010110011
0100100001	010110100
0101101000	010110101
0110110000	010110110
0111111001	010110111
1000111010	010111000
1001110011	010111001
1010101100	010111010
1010100101	010111011
1001011110	010111100

1000010110	010111101
0111001101	010111110
0110000100	010111111
0101000011	011000000
0100001010	011000001
0011010001	011000010
0010011000	011000011
0001100000	011000100
0000101001	011000101
0000110010	011000110
0001111011	011000111
0010111100	011001000
0011110101	011001001
0100101110	011001010
0101100110	011001011
0110011101	011001100
0111010100	011001101
1000001011	011001110
1001000010	011001111
1010000001	011010000
1010001000	011010001
1001010000	011010010
1000011001	011010011
0111100010	011010100
0110101011	011010101
0101110100	011010110
0100111101	011010111
0011111110	011011000
0010110110	011011001
0001101101	011011010
0000100100	011011011
0000011011	011011100
0001010010	011011101
0010001001	011011110
0011000000	011011111
0100000000	011100000
0101001001	011100001
0110010010	011100010
0111011011	011100011
1000100100	011100100
1001101101	011100101
1010110110	011100110
1010111110	011100111
1001111101	011101000
1000110100	011101001
0111101011	011101010
0110100010	011101011
0101011001	011101100
0100010000	011101101
0011001000	011101110
0010000001	011101111

0001000010	011110000
0000001011	011110001
0000010100	011110010
0001011101	011110011
0010100110	011110100
0011101110	011110101
0100110101	011110110
0101111100	011110111
0110111011	011111000
0111110010	011111001
1000101001	011111010
1001100000	011111011
1010011000	011111100
1010010001	011111101
1001001010	011111110
1000000011	011111111
0111000100	100000000
0110001101	100000001
0101010110	100000010
0100011110	100000011
0011100101	100000100
0010101100	100000101
0001110011	100000110
0000111010	100000111
0000111001	100001000
0001110000	100001001
0010101000	100001010
0011100001	100001011
0100011010	100001100
0101010011	100001101
0110001100	100001110
0111000101	100001111
1000000110	100010000
1001001110	100010001
1010010101	100010010
1010011100	100010011
1001100011	100010100
1000101010	100010101
0111110001	100010110
0110111000	100010111
0101111000	100011000
0100110001	100011001
0011101010	100011010
0010100011	100011011
0001011100	100011100
0000010101	100011101
0000001110	100011110
0001000110	100011111
0010000101	100100000
0011001100	100100001
0100010011	100100010

0101011010	100100011
0110100001	100100100
0111101000	100100101
1000110000	100100110
1001111001	100100111
1010111010	100101000
1010110011	100101001
1001101100	100101010
1000100101	100101011
0111011110	100101100
0110010110	100101101
0101001101	100101110
0100000100	100101111
0011000011	100110000
0010001010	100110001
0001010001	100110010
0000011000	100110011
0000100000	100110100
0001101001	100110101
0010110010	100110110
0011111011	100110111
0100111100	100111000
0101110101	100111001
0110101110	100111010
0111100110	100111011
1000011101	100111100
1001010100	100111101
1010001011	100111110
1010000010	100111111
1001000001	101000000
1000001000	101000001
0111010000	101000010
0110011001	101000011
0101100010	101000100
0100101011	101000101
0011110100	101000110
0010111101	101000111
0001111110	101001000
0000110110	101001001
0000101101	101001010
0001100100	101001011
0010011011	101001100
0011010010	101001101
0100001001	101001110
0101000000	101001111
0110000000	101010000
0111001001	101010001
1000010010	101010010
1001011011	101010011
1010100100	101010100
1010101101	101010101

1001110110	101010110
1000111110	101010111
0111111101	101011000
0110110100	101011001
0101101011	101011010
0100100010	101011011
0011011001	101011100
0010010000	101011101
0001001000	101011110
0000000001	101011111
0000000010	101100000
0001001011	101100001
0010010100	101100010
0011011101	101100011
0100100110	101100100
0101101110	101100101
0110110101	101100110
0111111100	101100111
1000111011	101101000
1001110010	101101001
1010101001	101101010
1010100000	101101011
1001011000	101101100
1000010001	101101101
0111001010	101101110
0110000011	101101111
0101000100	101110000
0100001101	101110001
0011010110	101110010
0010011110	101110011
0001100101	101110100
0000101100	101110101
0000110011	101110110
0001111010	101110111
0010111001	101111000
0011110000	101111001
0100101000	101111010
0101100001	101111011
0110011010	101111100
0111010011	101111101
1000001100	101111110
1001000101	101111111
1010000110	110000000
1010001110	110000001
1001010101	110000010
1000011100	110000011
0111100011	110000100
0110101010	110000101
0101110001	110000110
0100111000	110000111
0011111000	110001000

0010110001	110001001
0001101010	110001010
0000100011	110001011
0000011100	110001100
0001010101	110001101
0010001110	110001110
0011000110	110001111
0100000101	110010000
0101001100	110010001
0110010011	110010010
0111011010	110010011
1000100001	110010100
1001101000	110010101
1010110000	110010110
1010111001	110010111
1001111010	110011000
1000110011	110011001
0111101100	110011010
0110100101	110011011
0101011110	110011100
0100010110	110011101
0011001101	110011110
0010000100	110011111
0001000011	110100000
0000001010	110100001
0000010001	110100010
0001011000	110100011
0010100000	110100100
0011101001	110100101
0100110010	110100110
0101111011	110100111
0110111100	110101000
0111110101	110101001
1000101110	110101010
1001100110	110101011
1010011101	110101100
1010010100	110101101
1001001011	110101110
1000000010	110101111
0111000001	110110000
0110001000	110110001
0101010000	110110010
0100011001	110110011
0011100010	110110100
0010101011	110110101
0001110100	110110110
0000111101	110110111
0000111110	110111000
0001110110	110111001
0010101101	110111010
0011100100	110111011

0100011011	110111100
0101010010	110111101
0110001001	110111110
0111000000	110111111
1000000000	111000000
1001001001	111000001
1010010010	111000010
1010011011	111000011
1001100100	111000100
1000101101	111000101
0111110110	111000110
0110111110	111000111
0101111101	111001000
0100110100	111001001
0011101011	111001010
0010100010	111001011
0001011001	111001100
0000010000	111001101
0000001000	111001110
0001000001	111001111
0010000010	111010000
0011001011	111010001
0100010100	111010010
0101011101	111010011
0110100110	111010100
0111101110	111010101
1000110101	111010110
1001111100	111010111
1010111011	111011000
1010110010	111011001
1001101001	111011010
1000100000	111011011
0111011000	111011100
0110010001	111011101
0101001010	111011110
0100000011	111011111
0011000100	111100000
0010001101	111100001
0001010110	111100010
0000011110	111100011
0000100101	111100100
0001101100	111100101
0010110011	111100110
0011111010	111100111
0100111001	111101000
0101110000	111101001
110101000	111101010
0111100001	111101011
1000011010	111101100
1001010011	111101101
1010001100	111101110

1010000101	111101111
1001000110	111110000
1000001110	111110001
0111010101	111110010
0110011100	111110011
0101100011	111110100
0100101010	111110101
0011110001	111110110
0010111000	111110111
0001111000	111111000
0000110001	111111001
0000101010	111111010
0001100011	111111011
0010011100	111111100
0011010101	111111101
0100001110	111111110
0101000110	111111111

APPENDIX C

MATLAB PROGRAM TO GENERATE PLA TRUTH TABLE

%This M-file generates a truth table for a pla used in thesis

clear

% input Vectors

```
seed7=[0 0 0
        0 0 1
        0 1 0
        0 1 1
        1 0 0
        1 0 1
        1 1 0
        1 1 0
        1 0 1
        1 0 0
        0 1 1
        0 1 0
        0 0 1
        0 0 0];
```

```
seed8=[0 0 0
        0 0 1
        0 1 0
        0 1 1
        1 0 0
        1 0 1
        1 1 0
        1 1 1
        1 1 1
        1 1 0
        1 0 1
        1 0 0
        0 1 1
        0 1 0
        0 0 1
        0 0 0];
```

```

seed11=[0 0 0 0
        0 0 0 1
        0 0 1 0
        0 0 1 1
        0 1 0 0
        0 1 0 1
        0 1 1 0
        0 1 1 1
        1 0 0 0
        1 0 0 1
        1 0 1 0
        1 0 1 0
        1 0 0 1
        1 0 0 0
        0 1 1 1
        0 1 1 0
        0 1 0 1
        0 1 0 0
        0 0 1 1
        0 0 1 0
        0 0 0 1
        0 0 0 0];

```

```

sz7=max(size(seed7));
sz8=max(size(seed8));
sz11=max(size(seed11));

```

```

n7=ceil(512/sz7);
n8=ceil(512/sz8);
n11=ceil(512/sz11);

```

```

for j=1:n7

```

```

    input7=[input7;seed7];
end

```

```

for j=1:n8

```

```

    input8=[input8;seed8];
end

```

```

for j=1:n11

```

```

    input11=[input11;seed11];
end

```

```

input7=input7(1:512,:);
input8=input8(1:512,:);

```

```
input11=input11(1:512,:);
```

```
snsv=[input11 input8 input7];
```

%This produces a 9-bit binary table

```
seedb1=[zeros(1,1);ones(1,1)];  
seedb2=[zeros(2,1);ones(2,1)];  
seedb3=[zeros(4,1);ones(4,1)];  
seedb4=[zeros(8,1);ones(8,1)];  
seedb5=[zeros(16,1);ones(16,1)];  
seedb6=[zeros(32,1);ones(32,1)];  
seedb7=[zeros(64,1);ones(64,1)];  
seedb8=[zeros(128,1);ones(128,1)];  
seedb9=[zeros(256,1);ones(256,1)];
```

```
szb1=max(size(seedb1));  
szb2=max(size(seedb2));  
szb3=max(size(seedb3));  
szb4=max(size(seedb4));  
szb5=max(size(seedb5));  
szb6=max(size(seedb6));  
szb7=max(size(seedb7));  
szb8=max(size(seedb8));
```

```
nb1=ceil(512/szb1);  
nb2=ceil(512/szb2);  
nb3=ceil(512/szb3);  
nb4=ceil(512/szb4);  
nb5=ceil(512/szb5);  
nb6=ceil(512/szb6);  
nb7=ceil(512/szb7);  
nb8=ceil(512/szb8);
```

```
for j=1:nb1
```

```
    outputb1=[outputb1;seedb1];
```

```
end
```

```
for j=1:nb2
```

```
    outputb2=[outputb2;seedb2];
```

```
end
```

```
for j=1:nb3
```

```

        outputb3=[outputb3;seedb3];
end

for j=1:nb4
    outputb4=[outputb4;seedb4];
end

for j=1:nb5
    outputb5=[outputb5;seedb5];
end

for j=1:nb6
    outputb6=[outputb6;seedb6];
end

for j=1:nb7
    outputb7=[outputb7;seedb7];
end

for j=1:nb8
    outputb8=[outputb8;seedb8];
end

output=[seedb9 outputb8 outputb7 outputb6 outputb5 outputb4
outputb3 ...
        outputb2 outputb1];

s=3;
for j=1:512
    sbg(j)=s;
end
sbg=sbg';

platt=[snsv sbg output]

```

APPENDIX D
DIGITAL SIMULATION OF PLA

```
sun2:/home5/esparza/thesis/magic
% esim thespla3.sim
ESIM (V3.5 03/27/91)
7482 transistors, 562 nodes (521 pulled up)
sim> I
initialization took 1539 steps
sim> I
initialization took 0 steps
sim> w I10 I9 I8 I7 I6 I5 I4 I3 I2 I1 X9 X8 X7 X6 X5 X4 X3
X2 X1
sim> V I10 00000
sim> V I9 00001
sim> V I8 00110
sim> V I7 01010
sim> V I6 00001
sim> V I5 00110
sim> V I4 01010
sim> V I3 00001
sim> V I2 00110
sim> V I1 01010
sim> I
initialization took 0 steps
sim> G
>00000:I10
>00001:I9
>00110:I8
>01010:I7
>00001:I6
>00110:I5
>01010:I4
>00001:I3
>00110:I2
>01010:I1
>00000:X9
>00000:X8
>00000:X7
>00000:X6
>00000:X5
>00000:X4
>00001:X3
>00110:X2
>01010:X1
sim> Q
```

LIST OF REFERENCES

1. Coughlin, R.F., and Driscoll, F.F., Operational Amplifiers & Linear Integrated Circuits, pp. 401-418, 4th ed., Prentice-Hall, Inc., 1991.
2. Analog Devices, Inc., Analog-Digital Conversion Handbook, pp.211-220., Prentice-Hall, Inc., 1986.
3. Hnatek, E.R., A User's Handbook of D/A and A/D Converters, pp. 14-21., John Wiley & Sons, Inc., 1976.
4. Jaeger, R.C., "Tutorial: Analog Data Acquisition Technology, Part II-Analog-to-Digital Conversion," *IEEE Micro*, Vol. 2, No. 3, pp. 50-54, 1982.
5. Van Valburg, J., and Van de Plassche, J., "An 8-b 650MHz Folding ADC," *IEEE Journal of Solid-State Circuits*, Vol. 27, No. 12, 1992.
6. Van Der Plassche, R. J., and Van Der Grift, R. E. J., "A High Speed 7-bit A/D Converter," *IEEE Journal of Solid State Circuits*, Vol 14, pp. 938-943, 1979.
7. Arbel, A. and Kurz, R., "Fast ADC," *IEEE Transactions on Nuclear Science*, Vol. 22, pp. 446-451, 1975.
8. Fiedler, U. and Seitzer, D., "A High-Speed 8-bit A/D Converter Based on a Gray-Code Multiple Folding Circuit," *IEEE Journal of Solid State Circuits*, Vol. 14, pp. 547-551, 1979.
9. Van Der Grift, R. E. J., and Van Der Plassche, R. J., "A Monolithic 8-bit Video A/D Converter," *IEEE Journal of Solid State Circuits*, Vol. 19, pp. 374-378, 1984.
10. Van Der Grift, R. E. J., Rutten, I. W. J. M., and Van Der Veen, M., "An 8-bit Video ADC Incorporating Folding and Interpolation techniques," *IEEE Journal of Solid State Circuits*, Vol. 22, pp. 944-953, 1987.
11. Van Der Plassche, R. J., and Baltus, P., "An 8-bit 100 Mhz Full-Nyquist Analog-to-Digital Converter," *IEEE Journal of Solid State Circuits*, Vol 23, pp. 1334-1344, 1988.

12. Esparza, J. A., and Pace, P. E., "A Preprocessing Architecture for Resolution Enhancement in High Speed Analog-to-Digital Converters," 1993 *IEEE International Symposium on Circuits and Systems*, Vol. 2, pp 1140-1143, 1993.
13. Pace, P. E., and Styer, D., "High Resolution Encoding Process for an Integrated Optical Analog-to-Digital Converter," accepted for publication by *Optical Engineering*, November , 1993.

BIBLIOGRAPHY

Allen, P. E., and Holdberg, D. R., CMOS Analog Circuit Design, Holt, Rinehart, and Winston, Inc., 1987.

Geiger, R. L., Allen, P. E., and Strader, N. R., VLSI Design Techniques for Analog and Digital Circuits, McGraw-Hill, Inc., 1990.

Sedra, A. S., and Smith, K. C., Microelectronic Circuits, Holt, Rinehart, and Winston, Inc., 1991.

Weste, N., and Eshraghian, K., Principles of CMOS VLSI Design: A Systems Perspective, Addison-Wesley, Inc., 1985.

INITIAL DISTRIBUTION LIST

	No. Copies
1. Defense Technical Information Center Cameron Station Alexandria VA 22304-6145	2
2. Library, Code 052 Naval Postgraduate School Monterey CA 93943-5002	2
3. Chairman, Code EC Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, CA 93943-5121	1
4. Professor Phillip E. Pace, Code EC/Pc Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, CA 93943	3
5. Professor Douglas J. Fouts, Code EC/Fs Department of Electrical and Computer Engineering Code EC/FS Naval Postgraduate School Monterey, CA 93943	1
6. Director, Training and Education MCCDC, Code C46 1019 Elliot Road Quantico, VA 22134-5027	1
7. Department Chairman Department of Electrical Engineering U.S. Naval Academy Annapolis, MD 21412	1
8. Capt. Jorge A. Esparza, USMC 6 Sellers Road, Apartment D Annapolis, MD 21402	3
9. Ms. Socorro Esparza 902 E. Brown Street Santa Ana, CA 92701	1